

# XenAppおよびXenDesktop 7.6

Jan 19, 2017

## 新機能

長期サービスリリース (LTSR)

Feature Pack 3: Install and upgrade

Profile Management 3で解決された問題

既知の問題

このリリースに含まれていない機能

## システム要件

## 製品の技術概要

概念およびコンポーネント

Active Directory

フォールトトレランス

配信方法

リファレンスアーキテクチャ

設計ガイド

実装ガイド

## 新しい展開環境

インストールの準備

仮想化環境の準備 : VMware

仮想化環境の準備 : Microsoft System Center Virtual Machine Manager

Microsoft System Center Configuration Managerを使用するための準備

グラフィカルインターフェイスを使ったインストール

コマンドラインを使ったインストール

サイトの作成

スクリプトを使用したVirtual Delivery Agentのインストールと削除

マシンカタログ

デリバリーグループ

XenAppの公開アプリケーションと公開デスクトップ

VM Hosted Apps

VDIデスクトップ

リモートPCアクセス

App-V

ローカルアプリケーションアクセスとURLリダイレクト

サーバーVDI

コンポーネントの削除

## アップグレードと移行

環境のアップグレード

XenApp 6.xからの移行

XenDesktop 4からの移行

## セキュリティ

Citrix XenAppおよびXenDesktopのセキュリティの概要

セキュリティに関するベストプラクティスと考慮事項

管理権限の委任

スマートカード

SSL

## ポリシー

ポリシーの使用

ポリシーテンプレート

ポリシーの作成

ポリシーの比較、優先度、モデル作成、およびトラブルシューティング

デフォルトのポリシー設定

ポリシー設定リファレンス

## 印刷

印刷構成の例

ベストプラクティス、セキュリティに関する考慮事項、およびデフォルトの操作

印刷に関するポリシーと設定

プリンターのプロビジョニング

印刷環境の保守

## ライセンス管理

## 接続とリソース

### 接続リース

### 仮想IPおよび仮想ループバック

### セカンダリデータベースの場所

### Delivery Controller環境

Controllerの追加、削除、または移動、またはVDAの移動

Active Directory OUベースのController検出

### セッション管理

### Studioでの検索の使用

### IPv4/IPv6のサポート

### クライアントフォルダーのリダイレクト

### Personal vDisk

インストールとアップグレード

構成と管理

ツール

表示、メッセージ、およびトラブルシューティング

### ユーザープロファイル

### HDX

Thinwire互換モード

HDX 3D Pro

Flashリダイレクト

ホストからクライアントへのリダイレクト

WindowsデスクトップOSのためのGPUアクセラレーション

WindowsサーバーOSのためのGPUアクセラレーション

OpenGLソフトウェアアクセラレータ

オーディオ機能

ネットワークトラフィックの優先度

[USBとクライアント側ドライブの考慮事項](#)

## 監視

[Director](#)

[Session Recording](#)

[Personal vDisk](#)

[構成ログ](#)

[Monitor Service OData API](#)

## SDK

[XenDesktopの管理モデルについて](#)

[SDKの導入](#)

[PowerShellコマンドレットのヘルプ](#)

## FIPSの展開例

### サードパーティ製品についての通知

[Citrix SCOM Management Pack for XenApp and XenDesktop](#)

[ライセンスサーバー用Citrix SCOM Management Pack](#)

# Feature Pack 3

Jan 04, 2017

ここでは次のことについて説明します。

[長期サービスリリース \(LTSR\)](#)

[Feature Pack 3](#)

[Feature Pack 2](#)

[Feature Pack 1](#)

[XenApp/XenDesktop 7.6](#)

## Feature Pack 3

Feature Pack 3の新しく強化された機能は、次の通りです。

インストールおよびアップグレードの手順については、[「Feature Pack3コンポーネントのインストールとアップグレード」](#)を参照してください。

### 解決された問題

更新されたVDAスタンドアロンパッケージにより、次のものが追加されます：

- Windows Desktop OS用の標準のVDAでの**Windows 10のサポート**（Enterprise Edition）。

更新されたUniversal Print Serverパッケージにより、次のものが追加されます：

- **Windows Server 2012 R2のサポート**。

Feature Pack 3 VDAには、次の新しいHDXテクノロジー機能や強化された機能が実装され、更新されたグループポリシー管理パッケージを使って管理されます。

- **Thinwire互換モードへの拡張**。このモードは、リソース集約型のビデオコーデックの使用を必要とせずにシンプルなグラフィックインターフェイス（Microsoft ExcelやWordなど）の表示パフォーマンスを向上させる更新されたエンコーダーを使用します。詳しくは、「[Thinwire互換モード](#)」を参照してください。
- **署名デバイスおよび描画タブレットのサポート**。管理者は、デバイスでクライアント側のポーリングを有効にするポリシーを設定し、またそのポリシーの一部として最適化モードを設定できます。これにより、ユーザーはXenAppおよびXenDesktopセッション内においてWacom署名パッドやタブレットなどのデバイスを使いワイドエリアネットワークで効率的に作業を実行できます。詳しくは、「[USBデバイスのポリシー設定](#)」を参照してください。
- **ビデオフォールバック防止ポリシー**。管理者は、サーバー側およびクライアント側においてマルチメディアコンテンツのフェッチおよびレンダリングをより詳細に制御できます。たとえばユーザーのデバイスがローカルでビデオを再生できな

い場合に、管理者は新しい設定を使ってビデオをサーバー上で自動的にレンダリングするか、またはそれをせずにユーザーにメッセージを表示するかを制御できます。詳しくは、「[マルチメディアのポリシー設定](#)」および「[Flashリダイレクトのポリシー設定](#)」を参照してください。

- **Flashリダイレクトの最小バージョンチェック。** Receiver for WindowsおよびReceiver for Linuxを使用してVDAにアクセスしているクライアントデバイスを対象としたものです。これにより管理者は、Flashリダイレクトで必要とされる最小バージョンの指定についてより詳細に制御できるようになります。最小バージョンの指定方法について詳しくは、「[Flashリダイレクト](#)」を参照してください。
- **スマートカード仮想チャネルの向上。** Receiver for Windows 4.3によるスマートカードレンダリングプラグアンドプレイのサポート、仮想スマートカード（Microsoft Windowsトラステッドプラットフォームモジュールベース仮想スマートカード）、およびパフォーマンスの向上を含みます。スマートカードの展開について詳しくは、「[スマートカード](#)」およびMicrosoft Windows TechCenterの[Virtual Smart Card Overview](#)を参照してください。
- **Framehawk仮想ディスプレイチャネルのVDAとの統合。** Framehawk仮想ディスプレイチャネルはスタンドアロンVDAパッケージと統合されています。HDXテクノロジーとFramehawkを統合することで、高パケット損失や輻輳が生じたワイヤレス接続など、リモートワークスペースのユーザーが以前に経験した詳しくは、[HDX Framehawk virtual channel administrator guide](#)を参照してください。
- **HDX 3D Pro VDIに対する全画面アプリのサポート。** Virtual Delivery Agent (VDA) for HDX 3D Proは、単一モニターのICAセッションにおいて、3Dおよびゲームアプリを含む全画面アプリをサポートします。
- **無損失圧縮用のカスタムホットキー構成。** HDX 3D Proを使い、管理者はレジストリ設定を構成してカスタムキーボードショートカットを設定し、デフォルト（Alt+Shift+1）のショートカットキー設定を上書きできます。ユーザーは、カスタムホットキーを使って損失圧縮と視覚的無損失圧縮、または損失圧縮と無損失圧縮とを簡単に切り替えることができます。詳しくは、「[WindowsデスクトップOSのためのGPUアクセラレーション](#)」を参照してください。
- **Call Home。** このバージョンのCall Homeでは、管理者はPowerShellコマンドレット（`Start-TelemetryUpload`）を実行してVDA診断情報を抽出します。このコマンドレットを使い、サポート担当者へメールを送信するローカルZIPファイルを作成したり、あるいはSupport Service Request（SR）番号と組み合わせられた資格情報を使ってCitrix Insights Servicesに自動的に情報をアップロードしたりできます。これにより管理者とサポート担当者はVDA関連の問題を識別してトラブルシューティングを実施できます。PowerShellコマンドレットはFP3のVDAインストーラーにより提供されています。
- **強化されたHDXプリンターのサポートおよび構成。** プリントサーバーとVDA上の印刷サブシステムの常時ログ、およびCitrix Customer Experience Improvement Program（CEIP）へ参加するオプションがあります。詳しくは、「[コマンドラインを使ったインストール](#)」および「[ベストプラクティス、セキュリティに関する考慮事項、およびデフォルトの操作](#)」を参照してください。
- **更新されたポリシーテンプレート。** このFeature Packには、高サーバースケラビリティとWANの最適化のための新しいポリシーテンプレートが含まれ、高品位ユーザーエクスペリエンステンプレートは最高品位テンプレートと置き換えられています。詳しくは、「[ポリシーテンプレート](#)」を参照してください。
- **32ビットカーソルのサポート。** カスタムカーソルに依存する3Dアプリケーションでのユーザーエクスペリエンスを向上させます。
- **デスクトップコンポジションリダイレクト（DCR）。** DCRは、デフォルトで無効になっています。

更新されたStoreFrontパッケージにより、次のものが追加されます：

- Transport Layer Security (TLS) 1.0および1.1のサポート。Secure Sockets Layer (SSL) 3.0はサポートされず、Citrixはこれを使用しないことを強くお勧めします。

詳しくは、「[StoreFront](#)」のドキュメントを参照してください。

SUSEおよびRed HatディストリビューションをベースとしたLinux仮想デスクトップを作成できます。Linux仮想マシンを準備して新しいLinux VDAソフトウェアをインストールし、Delivery Controllerを構成したらStudioを使ってユーザーがデスクトップを使用できるようにします。詳しくは、次のドキュメントを参照してください。

 [SUSEのインストールガイド](#)

 [Red Hatのインストールガイド](#)

## Feature Pack 2

Feature Pack 2の新機能および強化された機能は、次の通りです。

StoreFrontには、次の新機能および強化された機能があります。

### 更新されたStoreFront管理コンソール

統合Receiverエクスペリエンスのサポート StoreFront 3.0により、Receiverユーザーに対するエンドユーザーアプリケーションとデスクトップ選択のカスタマイズおよびブランド化を集中管理します。所属する組織がインターフェイスをカスタマイズして、ロゴ、色、およびそのほかを変更できます。ReceiverとStoreFrontの新しい機能の一部に次のようなものがあります。

- **アプリケーション体型の簡素化。** 馴染みのある汎用的なアプリケーション選択方式を採用しています。
  - お勧めのアプリケーショングループ。管理者はアプリケーションを論理的にグループ化して構成し、アプリケーションの選択の画面でユーザーに示します。
  - アプリケーションのフォルダービュー。StoreFrontにWeb Interfaceの機能性を復活させています。
  - お気に入り。すばやくアクセスできるよう、アプリケーションをお気に入りに簡単に追加または削除できます。
- **サーバー管理型のユーザーエクスペリエンス。** アプリケーション選択のエクスペリエンスはサーバーで構成します。
  - 一度の変更であらゆる場所に展開。ユーザーのアプリケーション選択エクスペリエンスの変更は、StoreFrontサーバー上で管理者により実行されます。
  - ユーザーに対する首尾一貫したアプリケーション選択エクスペリエンス。Receiverを実行している任意のデバイスは、サーバーからHTML5としてアプリケーション選択エクスペリエンスを受け取り、ユーザーがスマートフォン、タブレット、およびデスクトップなどのフォームファクターを移動しても首尾一貫した選択エクスペリエンスを提供します。
  - Receiverの機能からアプリケーション選択エクスペリエンスを切り離し。エンドユーザートレーニングがいらない、またはユーザーエクスペリエンスに対してインパクトを与えないクライアントのアップグレード。レガシークライアントコストを必要としないレガシーユーザーエクスペリエンスをサポートします。
- **目的にかなったユーザーエクスペリエンスのカスタマイズオプション。**適切なカスタマイズレベルを選択します。
  - 追加設定不要なユーザーエクスペリエンスの更新。よりモダンなエンドユーザーアプリケーション選択画面の外観と操作性。
  - サーバーで構成するブランド設定のカスタマイズ。企業ロゴ、色、およびお勧めのアプリケーショングループをブランド化。

- より奥深いカスタマイズおよびブランド設定のオプション。CSSおよびスクリプトAPIを使用して、より深いレベルのカスタマイズを実現します。
- StoreFront管理コンソールを使って、次のReceiver関連タスクを実行します。
  - ストアのデフォルトとしてReceiver for Webを設定します。
  - Receiver for WebのWebサイトを作成します。
  - Webサイトの種類を変更します。
  - Webサイトの外観のカスタマイズ。

「[Citrix Receiver for Webサイトの管理](#)」を参照してください。

- **おすすめアプリのグループの管理** アプリケーショングループは、特定のカテゴリと関連した、またはそれと適合したアプリケーションのグループで、より簡単に検出できるようにします。 StudioおよびStoreFront管理 コンソールで、キーワード、アプリケーション名、またはカテゴリを使ってアプリケーショングループを定義できます。

「[お勧めのアプリケーションの作成および管理](#)」を参照してください。

## XMLサービススペースの認証

すべてを構成する場合、最初に発生したイベントによって未使用のセッションが終了します。 PowerShellコマンドレットを使って、この機能を有効または無効にします。 詳しくは、「[XMLサービススペース認証](#)」を参照してください。

HDX RealTime Optimization Packには次の新機能および強化された機能があります。

- Lync UIモードでのMicrosoft Skype for Businessクライアント、Microsoft Lync 2013クライアント、Microsoft Lync 2010クライアントのサポート。
- Lync Server 2013 Autodiscover Serviceのサポート。
- Lync 2010クライアントのコールパークおよびコールピックアップ。
- Lync 2010クライアントの着信の転送、および同時呼び出しの制御。
- Macのサポート。
- Kerberos認証。
- ダイヤルパッドのプラス (+) 記号。
- Microsoft Windows 10 Technical Previewのサポート。

詳しくは、「[HDX RealTime Optimization Pack 1.8 for Microsoft Lync](#)」を参照してください。

Session Recordingの新機能、および強化された機能は以下のとおりです。

- Session Recordingデータベースコンポーネントのインストール時にデータベース接続の資格情報を指定できるようになりました。
- Session RecordingデータベースコンポーネントとSession Recording Serverコンポーネントのインストール時にデータベースの接続を、Session Recording Agentコンポーネントのインストール時にSession Recording Serverの接続をテストできるようになりました。
- Microsoft Shared Management Objectsは、Session Recordingデータベースのインストールに不要となりました。
- このプログラムに参加すると、統計情報や使用状況が匿名でCitrixに送信されます。 詳しくは、「[Citrixカスタマーエクスペリエンス向上プログラム \(CEIP\) について](#)」を参照してください。

詳しくは、「[Session Recording - XenApp 7.6 FP1、FP2、およびLTSR](#)」を参照してください。



SUSEおよびRed HatディストリビューションをベースとしたLinux仮想デスクトップを作成できるようになりました。Linux仮想マシンを準備して新しいLinux VDAソフトウェアをインストールし、Delivery Controllerを構成したらStudioを使ってユーザーがデスクトップを使用できるようにします。詳細については、上記Feature Pack 3セクションの最新のドキュメントを参照してください。

FramehawkはCitrix HDXテクノロジーを拡張する新しいICA仮想チャネルで、仮想デスクトップおよびアプリケーションの高定義なユーザーエクスペリエンスを配信できます。

Framehawk仮想チャネルは、ブロードバンドワイヤレス接続上でユーザーへの仮想デスクトップとアプリケーションの配信を最適化します。これは、非可逆ネットワーク接続下のモバイルデバイスに最適です。HDXテクノロジーとFramehawkを統合することで、高パケット損失や輻輳が生じたワイヤレス接続など、リモートワークスペースのユーザーが以前に経験した不十分なインタラクティブ環境に対して、軽快で直観的なユーザーエクスペリエンスを提供します。

詳しくは、XenAppおよびXenDesktop 7.6 Feature Pack 3、ならびにFeature Pack 2の両方についての情報を含む次のドキュメントを参照してください。

 [Framehawk管理者ガイド](#)

このバージョンのDirectorはFramehawk仮想チャネルに対応しており、バージョン7.6.200で見つかった次の問題が修正されています。このコンポーネントのインストールについて詳しくは、[Directorのドキュメント](#)を参照してください。

[解決された問題](#)

## Feature Pack 1

Feature Pack 1の新機能および強化された機能は、次の通りです。

Session Recordingを使用すると、接続の種類を問わず、企業ポリシーおよび既定のコンプライアンスを遵守するXenAppを実行する任意のサーバー上で、画面上のアクティビティを録画することができます。Session Recordingにより録画、カタログ化、およびアーカイブされるセッションを、後で取得して再生することができます。詳しくは、「[Session Recording - XenApp 7.6 FP1、FP2、およびLTSR](#)」を参照してください。

Director 7.6.200では、[ユーザーの詳細] ビューや [マシン詳細] ビューでSession Recordingポリシーを有効または無効にすることができます。Directorコンソールを使用して、Session Recordingポリシーを作成およびアクティブ化します。

Session RecordingとDirectorの統合について詳しくは、「」を参照してください。詳しくは、[Session Recording - XenApp 7.6 FP1、FP2、およびLTSR](#)」を参照してください。

CitrixライセンスサーバーCustomer Experience Improvement Program (CEIP) およびCall Home — 匿名または記名により構成、パフォーマンス、エラー、および使用データがCitrix製品により展開環境から収集され、そのデータを自動的にCitrixに送

信する任意のデータ収集プログラムです。詳しくは、「[CitrixライセンスサーバーCustomer Experience Improvement Program \(CEIP\) およびCall Home](#)」を参照してください。ライセンスサーバーVPXではサポートされません。

Citrixカスタマーエクスペリエンス向上プログラム (CEIP) では、お客様の製品の使用状況データをCitrix製品のデザインおよび開発に役立たせていただきます。このプログラムにご参加いただくと、お客様のCitrix製品環境に関する情報が匿名でCitrixに送信されます。お客様からの情報は、Citrix製品の品質、信頼性、およびパフォーマンスを向上させる目的で使用させていただきます。

このプログラムについて詳しくは、「[Citrixカスタマーエクスペリエンス向上プログラムについて](#)」を参照してください。

Citrix HDX RealTime Optimization Pack 1.7 for Microsoft Lyncでは、XenDesktopおよびXenApp環境でMicrosoft Lyncのリアルタイムオーディオ/ビデオ会議やUSBまたはボイスオーバーIPの企業向け電話に対応する高度にスケーラブルなソリューションをLinuxおよびWindowsデバイスのユーザーに提供します。このOptimization Packは既存のMicrosoft Lyncインフラストラクチャを活用し、デバイスでネイティブに動作するほかのMicrosoft Lyncエンドポイントと連動します。

詳しくは、「[HDX RealTime Optimization Pack 1.7 for Microsoft Lync](#)」を参照してください。

## XenApp 7.6およびXenDesktop 7.6

この製品リリースでの新機能、および強化された機能は以下のとおりです。

セッションの事前起動機能とセッション残留機能を使用すると、セッションが要求される前にセッションを開始したり（セッションの事前起動）、ユーザーがすべてのアプリケーションを閉じた後もアプリケーションセッションをアクティブな状態で保持したり（セッション残留）できます。これにより、ユーザーがアプリケーションにすばやくアクセスできるようになります。これらの機能はサーバーOSマシンでのみサポートされます。

デフォルトでは、セッションの事前起動とセッション残留は無効になっています。セッションはユーザーがアプリケーションを開始すると開始（起動）され、セッションで開いていた最後のアプリケーションを閉じるまでアクティブな状態で保持されます。これらの機能は、デリバリーグループのすべてのユーザーについて有効にしたり、特定のユーザーについてのみ有効にしたりできます。

ユーザーがアプリケーションを起動しない場合に、使用されないセッションをどのくらい保持するかを指定するには、タイムアウトおよびサーバー負荷の2つのしきい値を構成します。タイムアウトおよびサーバー負荷のしきい値をすべて設定することができます。この場合、最初に発生したイベントによって事前起動セッションや残留セッションが終了します。

詳しくは、「[セッションの事前起動およびセッション残留の構成](#)」を参照してください。

サーバーOSマシン用のデリバリーグループの作成または編集時に、StoreFrontまたはCitrix Receiverでの認証が不要な匿名アクセスを許可できます。たとえば、キオスクからアプリケーションにアクセスする場合、アプリケーション自体での認証が必要な場合がありますが、Citrixのアクセスポータルやツールでは資格情報が要求されません。

既存のデリバリーグループを編集して、特定のユーザー、認証が不要なユーザー、またはその両方のアクセスを許可できます。認証が不要なユーザー（匿名ユーザー）のアクセスを許可する場合は、認証が不要なStoreFrontストアを作成しておく必要があります。

詳しくは、「[Users](#)」を参照してください。

サイトデータベースを常に使用可能状態にするために、Microsoft社の高可用性ベストプラクティスに従って耐障害性の高いSQL Server展開から開始することをお勧めします。ただし、ネットワークの問題および中断によってDelivery Controllerがデータベースにアクセスできなくなり、その結果、ユーザーがアプリケーションやデスクトップに接続できなくなる場合があります。

SQL Server高可用性ベストプラクティスを補完する接続リース機能を使用すると、サイトデータベースが使用不可の場合でも、ユーザーが直近で使用したアプリケーションやデスクトップに接続および再接続できるようになります。

通常、多くの公開リソースを使用できるユーザーでも、定期的に使用するのはその一部のみです。接続リース機能を有効にすると、通常時（データベースが使用可能なとき）に、最近使用されたアプリケーションやデスクトップのユーザー接続が各Controllerでキャッシュされます。データベースを使用できなくなると、Controllerはリース接続モードになります。この場合、ユーザーがStoreFrontから最近使用したアプリケーションやデスクトップへの接続を試みると、キャッシュされている操作が「再生」されます。

詳しくは、「[接続リース](#)」を参照してください。

アプリケーションをフォルダーで分類できるため、Studioで多数のアプリケーションを容易に管理できます。

デリバリーグループのアプリケーションは、デフォルトで単一フォルダー内に表示されます。Studioの [デリバリーグループ] ノードでフォルダーを追加して、そこに既存のアプリケーションを移動できます。これらのフォルダーは、ドラッグアンドドロップで移動または入れ子にしたり、名前を変更したりできます。これらの操作は、[操作] メニューからも実行できます。また、デリバリーグループに新しいアプリケーションを追加するときにフォルダーを作成して、そこにそのアプリケーションを追加することもできます。

詳しくは、「[アプリケーションフォルダーの管理](#)」を参照してください。

XenApp 6.5の移行プロセスでは、XenApp 6.5ファームをXenApp 7.6以降のサイトにすばやく効率的に移行できます。この移行方法では、多数のアプリケーションとCitrixグループポリシーを含む環境で、アプリケーションとCitrixグループポリシーを手動で移行する場合のエラーの発生リスクを軽減させることができます。

XenApp 7.6のコアコンポーネントをインストールしてサイトを作成したら、次の手順で移行プロセスを実行します。

- 各XenApp 6.5ワーカー上でXenApp 7.6インストーラーを実行します。これにより、Virtual Delivery Agent for Windows Server OSにアップグレードされます。
- いずれかのXenApp 6.5コントローラー上でPowerShellエクスポートコマンドレットを実行して、アプリケーション設定とCitrixポリシー設定をXMLファイルにエクスポートします。
- 必要に応じてXMLファイルを編集して、新しいサイトにインポートしないデータや設定を削除します。XMLファイルをカスタマイズすることにより、ポリシー設定とアプリケーション設定を段階的にXenApp 7.6サイトにインポートできます（一部を先にインポートしてから残りを後でインポートするなど）。
- 新しいXenApp 7.6 Controller上でPowerShellインポートコマンドレットを実行して、XMLファイルから新しいXenAppサイトに設定をインポートします。
- 新しいサイトを必要に応じて再構成してテストします。

詳しくは、「[XenApp 6.xからの移行](#)」を参照してください。

Citrixカスタマーエクスペリエンス向上プログラム (CEIP) では、お客様の製品の使用状況データをCitrix製品のデザインおよび開発に役立たせていただきます。このプログラムにご参加いただくと、お客様のCitrix製品環境に関する情報が匿名でCitrixに送信されます。お客様からの情報は、Citrix製品の品質、信頼性、およびパフォーマンスを向上させる目的で使用させていただきます。

このプログラムには、サイトを作成またはアップグレードした後で簡単にご参加いただけます。このプログラムへの参加およびその解除はいつでも自由に行えます。Studioのナビゲーションペインで [構成] を選択し、[製品サポート] タブに表示される手順に従ってください。

このプログラムについて詳しくは、[『Citrixカスタマーエクスペリエンス向上プログラムについて』](#) を参照してください。

パフォーマンスへの影響を軽減させるため、同時操作、Personal vDiskストレージインベントリの同時更新、およびホスト接続で同時に実行できる操作の最大数を指定できるようになりました。

詳しくは、「[接続の編集](#)」を参照してください。

PvDイメージの更新時に、Studioに状態およびエラーに関するより詳細なレポートが表示されます。また、Studioの [ライセンス] ノードに包括的なライセンスアラートが表示されます。

VDAがインストールされているマシンとVDAのデリバリーグループでSSL/TLSを構成して、ユーザーとVDA間でSSL/TLS接続を有効にできます。

詳しくは、「[SSL](#)」を参照してください。

公開アプリケーションでは、Windows Server 2008 R2およびWindows Server 2012 R2を実行するマシンでMicrosoft社の仮想IP機能を有効にして使用できます。また、仮想ループバックを管理する、新しいCitrixポリシー設定を追加できます。優先ループバック機能も使用できます。

詳しくは、「[仮想IPおよび仮想ループバック](#)」を参照してください。

ホストのクリップボードからクライアントへのまたはクライアントのクリップボードからユーザーセッションへのデータの共有を制限または有効にすることができます。共有が制限される場合は、それぞれのケースで共有できるデータ形式を指定できます。

詳しくは、「[ICAのポリシー設定](#)」を参照してください。

リモートユーザーの許可なしに、ローカルユーザーによってリモートセッションが切断されるのを回避できるようになりました。

リモートセッションを切断する場合は、マウスを動かすか、またはキーボードのキーを押すとローカルモニターが起動します (以前のリリースでは、Ctrl+Alt+Delキーを2回押すと、ログオン画面が表示されていました)。

解像度の低いデバイスから接続した後に解像度の高いデバイスに戻ったときに、アイコンの位置が保持されるようになりました。

このリリースでは、最適化された仮想チャネルを使用できない特殊なUSBデバイスに対して汎用USBリダイレクト機能が提供されます。この機能では、任意のUSBデバイスをクライアントマシンから仮想デスクトップにリダイレクトします。これにより、ユーザーはさまざまな種類の汎用USBデバイスを、デスクトップセッションに物理的に接続されているかのように使用できます。

以下の考慮事項があります。

- ユーザーデバイスにデバイスドライバーをインストールする必要はありません
- USBクライアントドライバーはVDAマシン上にインストールされます

この機能を使用するにはWindows Server 2012 R2が必要です。また、既存のバージョンのReceiver for WindowsでアクセスするRDSホスト上の公開デスクトップ（非ダブルホップ）がサポートされます。この機能を使用する場合、ホスト上にUSBクライアントドライバーがインストールされます。このため、これらのドライバーにRDSH for Windows 2012 R2プラットフォームとの互換性がある必要があります。

Director 7.6.100での新機能、および強化された機能は以下のとおりです。ダウンロード、インストール、およびアップグレードについては、<http://support.citrix.com/article/CTX200330>を参照してください。

- **仮想マシンの使用データ。** サイトの仮想マシンの使用状況がリアルタイムで表示されるため、管理者はサイトの処理能力が低下したときにすばやく対応できます。仮想マシンの使用データは、デスクトップOSおよびサーバーOSの可用性に基づいて分類されます。
  - **デスクトップOSの可用性。** デスクトップOSマシン（VDI）の現在の状態をサイト全体または特定のデリバリーグループについて可用性に基づいて表示します。
  - **サーバーOSの可用性。** サーバーOSマシンの現在の状態をサイト全体または特定のデリバリーグループについて可用性に基づいて表示します。
- **エクスポート機能の拡張。** 傾向レポートをCSV、PDF、およびExcel形式でエクスポートできるようになりました。
- **ズームインドリルダウン機能。** ドリルダウン機能は、Director 7.6で追加されました。この機能が拡張され、傾向グラフの特定の時間範囲をクリックして拡大して、その傾向の詳細情報を確認できるようになりました。これにより、その傾向データの影響を受けているユーザーやリソースを把握できます。

ユーザー接続に影響する可能性のあるライセンスの問題が発生すると、ライセンスアラートが表示されます。また、その問題を解決するための推奨タスクについても表示されます。たとえば、以下のアラートがDirectorに表示されます。

- すべてのライセンスの有効期限が切れた。
- ライセンスの有効期限が迫っている。
- Citrixライセンスの猶予期間が終了した。
- 追加猶予期間が有効で、すべてのライセンスが使用されている。

**ホストされたアプリケーションの使用量の表示。** 特定のデリバリーグループおよび期間について、最大同時使用量を示すグラフとアプリケーションごとの使用量を示す表が表示されます。[アプリケーションごとの使用量]の表では、特定のアプリケーションについての詳細や、そのアプリケーションを使用しているユーザー、および使用していたユーザーの情報を表すことができます。

**Hotfixの監視。** 管理者は、特定のマシンのVDA（物理または仮想）にインストールされているHotfixを [ユーザーの詳細] または [マシンの詳細] ビューで確認できます。

**フィルター機能の拡張。** フィルターデータをクリックして [ユーザーの詳細] ビュー、 [マシンの詳細] ビュー、 [エンドポイント詳細] ビュー、 および匿名セッションの情報を表示できます。

**DirectorとXenApp 6.5との互換性。** DirectorでXenApp 6.5の展開環境を監視できます。

詳しくは、「[Director](#)」を参照してください。

Citrix AppDNAの高速な分析機能と自動化されたアプリケーションの修復やパッケージ化により、デスクトップやWebアプリケーションを新しい環境用に移行したり変換したりする管理者の作業を支援し、日常的なアプリケーション更新の管理に役立ちます。AppDNA 7.6には、アプリケーションが同一オペレーティングシステムファミリの追加ビルドで機能するかどうかをテストする、新しいビルド評価ソリューションが含まれています。AppDNA 7.6の分析機能の強化により、必要なアプリケーション、アプリケーションフレームワーク、およびファイルが存在するかどうか、有効なGPOによって問題が発生するかどうか、およびWebアプリケーションがCitrix WorxWebと互換性があるかどうかを示されるようになりました。

詳しくは、「[AppDNA 7.6](#)」を参照してください。

**管理コンソールでのストア構成の簡素化。** StoreFront管理コンソールが更新され、以下の機能についてのStoreFront構成が簡素化されました。

- ユーザーサブスクリプション
- Receiver for Webのセッションのタイムアウトの設定
- ログオンページでのドメイン一覧の表示

**Receiver for Webでのフォルダービュー 認証不要なストアおよび必須ストアでアプリケーションがフォルダー階層で表示されるようになり、フォルダーパスの情報も表示されます。** これにより、ユーザーがWeb InterfaceからReceiver for Webに移行しやすくなりました。

**XenApp 6.5のKerberos制約付き委任。** StoreFrontでKerberos制約付き委任を使用するとパススルー認証が有効になり、WindowsやReceiverを実行するデバイスが不要になります。

**単一完全修飾ドメイン名 (FQDN) によるアクセス。** 単一FQDNでのアクセスをネットワーク内部および外部のユーザーに提供できます。

**XenApp Servicesサポートでのスマートカード認証。** StoreFrontサーバーでXenApp Servicesサポートサイトへのスマートカードによる認証が可能になり、Receiverやオペレーティングシステムに対する要件もありません。

**Receiver for Android、iOS、およびLinuxでのスマートカード認証。** Receiverの新しいバージョンでは、ローカルおよびリモートでスマートカードを使用してアプリケーションやデスクトップにアクセスできます。

**拡張可能な認証。** StoreFrontのフォームベースの認証を拡張する単一のカスタマイズポイントが提供されます。WorxHomeおよびReceiver for Webでは、XenMobileやXenApp/XenDesktopでの内部（直接）アクセスと外部（NetScaler Gateway経由）アクセスの両方でこの認証が使用されます。

詳しくは、「[StoreFront 2.6](#)」を参照してください。

Citrix Connector 7.5により、Microsoft System Center Configuration Manager (Configuration Manager) と、XenAppまたはXenDesktopが連携して動作するようになるため、Configuration Managerの使用をCitrix環境にまで拡張することができます。Citrix Connector 7.5では、XenApp 7.6およびXenDesktop 7.6のPlatinum Editionがサポートされています。

詳しくは、「[Citrix Connector 7.5 for System Center Configuration Manager 2012](#)」を参照してください。

Receiver for Chromeを使用すると、Google Chromeオペレーティングシステムが動作するデバイスから仮想デスクトップやホストされるアプリケーションにアクセスできます。ユーザーがReceiver for Chromeを使ってこれらのリソースにアクセスすると、デスクトップやアプリケーションが単一のウィンドウに表示されます。

Receiver for HTML5はStoreFrontサーバー上でホストされ、ユーザーは仮想デスクトップおよびホストされたアプリケーションにWebブラウザでアクセスします。ユーザーのデバイスにCitrix Receiverをローカルにインストールすることなく、デスクトップとアプリケーションにWebブラウザでアクセスできます。

このリリースでは、これらの両方のReceiverが、ホストされたアプリケーションまたは仮想デスクトップで実行されているアプリケーションのドキュメントをPDFに変換し、ローカルデバイスで表示したりローカル接続のプリンターで印刷したりする機能、強化されたクリップボード操作、エンドユーザーエクスペリエンス測定、およびホストされたアプリケーションのライセンス使用状況を追跡する機能を備えています。

詳しくは、「[Receiver for Chrome 1.4](#)」および「[Receiver for HTML5 1.4](#)」を参照してください。

HDX Real-Time Optimization Pack 1.5 for Microsoft Lyncでは、Lync対応のUSB電話、Lync 2010クライアントとLync Server 2013の混在構成、および非同期アップグレードをサポートしています。

詳しくは、「[HDX Real-Time Optimization Pack 1.5 for Microsoft Lync](#)」を参照してください。

# Feature Pack 3: Install and upgrade

Jan 04, 2017

このアーティクルは、XenAppおよびXenDesktop 7.6 FP3のすべてのコンポーネントについてのインストール手順について（何をどこにインストールするかなど）簡単に説明します。詳細な手順については、「[システム要件](#)」および製品のアーティクルリンクを参照してください。

FP3コンポーネントを[citrix.com](http://citrix.com)からダウンロードします。入手可能な内容については、現在の登録内容により異なります。

VDAをインストールするマシンに**VDAServerSetup.exe**をダウンロードして実行します。グラフィカルインターフェイスまたはコマンドラインを使用します。

詳しくは、「[スタンドアロンパッケージによるVDAのインストール](#)」を参照してください。

Windows 8.xまたはWindows 7を実行しているマシン上にインストールされたVDAをWindows 10にアップグレードするには、Windows 7およびWindows 8.xマシンをWindows 10に再イメージ化し、スタンドアロンパッケージを使ってWindows 10用にサポートされているVDAをインストールします。

詳しくは、「[環境のアップグレード](#)」を参照してください。

VDAをインストールするマシンに**VDAServerSetup.exe**をダウンロードして実行します。グラフィカルインターフェイスまたはコマンドラインを使用します。

VDA for Windows Server OSのインストールにより、2008および2010ランタイム（32ビットおよび64ビット）に加えて、Microsoft Visual C++ 2013ランタイム（32ビットおよび64ビット）が自動的に展開されます。Microsoft Visual C++ 2005は今後展開されません。これらの前提条件はサーバーの再起動を初期化し、再起動後に続いてVDAをインストールします。

詳しくは、「[スタンドアロンパッケージによるVDAのインストール](#)」を参照してください。

Feature Pack 3 VDAには、このリリースの新しいHDXテクノロジー機能や強化された機能が実装され、更新されたグループポリシー管理パッケージを使って管理されます。

Citrixポリシーを管理するサーバーで、**CitrixGroupPolicyManagement\_x64.msi**または**CitrixGroupPolicyManagement\_x86.msi**をダウンロードして実行します。次にStudioまたはGPMCを起動して、新しいポリシーや更新されたポリシーが表示されます。

更新されたポリシーについて詳しくは、以下を参照してください。拡張Thinwire互換モードについては、「[視覚表示のポリシー設定](#)」。署名デバイスと描画タブレットのサポートについては、「[USBデバイスのポリシー設定](#)」。ビデオフォールバックの防止については、「[Flashリダイレクト](#)」および「[マルチメディアポリシー設定](#)」。HDX Framehawk仮想チャネルについては、「[Framehawkポリシー設定](#)」。

Active DirectoryのマシンのグループのVirtual Delivery Agent (VDA) をインストール、アップグレード、または削除するスクリプト (**InstallVDA.bat** および **UninstallVDA.bat**) を含みます。使用する前にスクリプトを繻子精する必要があります。



詳しくは、「[Active Directoryのスク립トを使ったVirtual Delivery Agentのインストールまたは削除](#)」を参照してください。

UPSパッケージには、スタンドアロンUPSサーバーコンポーネント (**UpsServer\_x64.msi**および**UpsServer\_x86.msi**) および前提条件 (**vcredist\_x64.exe**、**vcredist\_x86.exe**、**cdf\_x64.msi**、および**cdf\_x86.msi**) の更新バージョンが含まれています。

Windows Server 2012 R2またはWindows Server 2012プリントサーバーで、コンポーネントのMSIであるUpsServer\_x64.msiまたはUpsServer\_x86.msiを展開して起動し、UPSサーバーコンポーネントをインストールします。UPSサーバーコンポーネントのインストール後には再起動する必要があります。

セッションネットワークプリンターをプロビジョニングするXenAppおよびXenDesktopホスト上にインストールするUPClientコンポーネントは、VDAインストールの一部です。

XenApp 6.5など、UPClientコンポーネントを別個に展開する環境の場合には追加手順が必要です。

FP3のすべてのこれらの構成については、[プリンターのプロビジョニング](#)を参照してください。

Windows 2008 32ビットプラットフォーム上へのUpsServer\_x86.msiを展開する場合にも、追加手順が必要です。詳しくは、「[コマンドラインを使ったインストール](#)」を参照してください。

IISで必要なすべての機能が選択されているかチェックしてください。完全な一覧については、[CTX142260](#)を参照してください。Directorを実行するサーバーに**DesktopDirector.msi**をダウンロードして実行します。CitrixGroupPolicyManagement.msiをインストールしていない場合はこれをインストールします。

パッケージに含まれているWMIProxy MSIファイルを使ってVDAにWMIProxyをインストールまたはアップグレードします。

パッケージに含まれているXDPOshSnapin MSIを使ってDelivery ControllerのXDPOshSnapin\_Hotfixをアップグレードします。これは、委任管理者またはカスタム管理者がDirectorでFramehawk仮想チャネル情報を見るのに必要です。

インストールと構成の完全な手順については、「[Director](#)」を参照してください。

AppDNA 7.6.5には新しいWindows 10レポートが含まれており、これによってWindows XP以降のアプリケーションがWindows 10で実行されるかどうかを把握することができます。インストーラーの**Citrix-AppDNA.msi**をダウンロードして、仮想マシンで実行します。すべての種類のライセンスで、インストーラーによって、初回インストールおよびアップグレードインストールが処理されます。試用版のライセンスの場合、Microsoft SQL Server Expressと連動するデータベースがインストーラーによって作成されます。

詳しくは、「[AppDNAのインストール](#)」を参照してください。

StoreFrontサーバーで管理者として**CitrixStoreFront-x64.exe**をダウンロードして実行します。StoreFront 3.0.1にはCitrix Receiver for HTML5 1.8が含まれています。

詳しくは、「[インストール、セットアップ、およびアンインストール](#)」を参照してください。

アップグレードするには、StoreFront展開へのアクセスを無効にして、StoreFrontサーバーを再起動します。StoreFrontサーバーで管理者として**CitrixStoreFront-x64.exe**をダウンロードして実行します。

詳しくは、「[アップグレード](#)」を参照してください。

Citrix Receiver for HTML5はStoreFrontでホストされ、デフォルトでStoreFrontと共にインストールされます。StoreFront 3.0.1にはCitrix Receiver for HTML5 1.8が含まれています。以前のバージョンのStoreFrontを実行していて、このバージョンのCitrix Receiver for HTML5をインストールまたはこれにアップグレードしたい場合は、StoreFront 2.5以降にReceiver for HTML5 1.8 ([CitrixHTML5Client-x64.exe](#)) をダウンロードして実行できます。

詳しくは、「[Citrix Receiver for HTML5の構成](#)」を参照してください。

Receiver for Windows 4.3.100は、FP3と共に配信されるVDA 7.6.300パッケージに含まれています。以前のバージョンのVDA上のReceiverのアップグレードまたはユーザーデバイス上でのインストールやアップグレードには、Citrix Receiver for Windows 4.3.100 ([CitrixReceiver.exe](#)) を使用します。

Citrix Lync Optimization Packがインストール済みのエンドポイントでは、これをアンインストールしてからCitrix Receiver for Windowsをアップグレードして、その後でOptimization Packをインストールし直してください。[CTX200340](#)を参照してください。

詳しくは、「[インストール](#)」を参照してください。

Linux仮想マシンを準備して新しい[Linux VDAソフトウェア](#)をインストールし、Delivery Controllerを構成したらStudioを使ってユーザーがデスクトップを使用できるようにします。

詳しくは、「[Linux Virtual Desktop Installation Guide for SUSE](#)」および「[Linux Virtual Desktop Installation Guide for Red Hat](#)」を参照してください。

これは、XenServer 6.5.0 Service Pack 1を実行しているユーザー用のHotfixです。このHotfixにより、スタンドアロンWindowsドライバー用のWindows 10証明が提供されます。

詳しくは、[CTX201974](#)を参照してください。

# Feature Pack 3で解決された問題

Jan 11, 2017

- Directorのアクティビティマネージャーで、一部のユーザーに対して開かれているアプリケーションが表示されないことがありました。  
[#LC2325]
- 複数のDirectorサイトを構成するとサイト名の一部が削除されることがありました。  
[#LC2967]

## Universal Print Serverのクライアント

### Universal Print Server

#### ユニバーサルプリンタードライバーのサポート

- ネットワークプリンターを使用中、印刷プレビューでOfficeドキュメントの表示が遅くなることがありました。  
[#LC1205]
- XenApp 6.5 Hotfix Rollup Pack 4およびUniversal Print Client 7.6がインストールされているサーバーで、ユーザーデバイスのセッションが応答しなくなることがありました。  
[#LC1562]
- この機能拡張は、Universal Print ServerおよびUniversal Print Clientのパフォーマンスカウンターを実装します。  
[#LC1820]
- Adobe Reader X1から印刷しようとする、印刷ダイアログの表示に最大20秒かかることがありました。  
[#LC2412]
- Citrix Print Manager Service (CpSvc.exe) プロセスが異常終了することがありました。この問題は、pServerNameの値が正しくない場合に発生します。  
[#LC2716]
- Universal Print ServerがVDAにインストールされていても、Citrixポリシーで無効になっている場合、PowerShellがVDAでプリンターコマンドを取得できないことがありました。  
[#LC2966]
- 印刷スプーラーサービスで、UpProv.dllの致命的な例外が断続的に発生することがありました。  
[#LC3055]

#### Universal Print Serverのクライアント

- この修正は、基本コンポーネント内のメモリ問題に対応しています。

[#LC1381]

- この機能拡張は、Universal Print ServerおよびUniversal Print Clientのパフォーマンスカウンターを実装します。

[#LC1820]

- この修正は、基本コンポーネント内のメモリ問題に対応しています。

[#LC2764]

XenAppおよびXenDesktopの7.6 Feature Pack 3 (7.6.300) と7.6 LTSRに含まれているVDA for Desktop OSは同じものです。XenAppおよびXenDesktop 7.7では、バージョン7.6.300と同じ修正が行われており、さらに互換性が向上しています。

Broker Agent	Remote PC
HDX 3d Pro	シームレスウィンドウ
HDX MediaStream Flashリダイレクト	セッション/接続
HDX MediaStream Windows Mediaリダイレクト	サイト/ファームの管理
インストール、アンインストール、アップグレード	スマートカード
キーボード/マウス	システムの例外
ローカルアプリケーションアクセス	ユーザーエクスペリエンス
ログオン/認証	ユーザーインターフェイス
印刷	その他

## Broker Agent

- VDAの登録時に初期化中の状態で停止することがありました。この問題は、Citrix Desktop Serviceが数日間継続して実行されている場合に発生します。

[#LC0570]

- 「CName」機能を有効にすると、VDAの登録に時間がかかることがありました。登録時間を短縮するには、この修正を適用してから「CNameSuppressOrgFqdnLookup」値を有効にしてください。重要：「CNameSuppressOrgFqdnLookup」を有効にする場合は、「CName」機能も有効にする必要があります。

重要：「CNameSuppressOrgFqdnLookup」を有効にした場合、「CName」も有効にする必要があります。

この修正を有効にするには、以下のレジストリキーを作成する必要があります。

- HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\VirtualDesktopAgent  
値の名前：UseCnameLookup  
種類：REG\_DWORD  
値：有効= 1 (無効= 0)
- HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\VirtualDesktopAgent  
値の名前：CNameSuppressOrgFqdnLookup  
種類：REG\_DWORD  
値：有効= 1 (無効= 0)

[#LC0791]

- VDI-in-a-Boxモードが有効な場合、ユーザーセッションを介してVDAに再接続しようとするとう失敗することがあります。

[#LC1522]

- バージョン7.6のVDAがXenDesktop 5.6サイトのメンバーの場合、ユーザーが接続し、切断し、その後でもう一度接続しようとうすると、再接続に失敗し、VDAが登録解除されます。

[#LC1859]

## HDX 3D Pro

- HDX 3D Proが有効なデスクトップセッションでユーザーがカスタムのカーソルをロードした場合に、カーソルの色が正しく表示されなくなることがあります。

[#LC0917]

- デスクトップがHDX 3D Pro対応で簡体中国語版のWindowsを実行しているサードパーティ製CADアプリケーションのツールボックスおよび描画領域の上にマウスカーソルを移動すると、マウスカーソルが見えなくなることがあります。

[#LC1321]

- Citrix Receiverセッションを切断した後は、NVIDIAグラフィックカードがついたHDX 3D Proをインストールしたコンピューター上ではロックされたVDAセッションへの再接続は失敗します。

[#LC1629]

- 測定値150% DPIのデュアルモニターHDX 3D Pro VDAセッションでは、マウスポインターおよびそのほかの画面アートエフェクトはVDAセッションの外側に表示され、そのためアクセスできないことがあります。

[#LC2502]

- ハードウェアアクセラレーションが有効な場合、CD Viewerウィンドウのサイズを変更した後、ReceiverセッションでAutoCAD 2011が失敗します。

[#LC2531]

- この修正により、HDXネットワークトラフィックの帯域幅キャップは20メガビット/秒 (Mbps) です。

[#LC2780]

- VDAが高レイテンシWANネットワークに接続されると、Nvidia Grid K1グラフィックボードを備えたHDX 3D Pro環境内のVDAセッションではグラフィックがぼやけて表示されることがあります。

[#LC2877]

- HDX 3D Proを有効にしたVDA上で、無損失圧縮が適用されているのかを示すインジケータータスクバーアイコン (LLIndicator.exe) が無損失圧縮状態を示すことができず、右クリックしても応答しなくなることがありました。

[#LC2931]

- HDX 3D Proデスクトップ上でユーザーがセッションを数回ロックまたはロック解除した後、マウスカーソルが非表示となることがありました。

[#LC3188]

- Desktop Viewerを全画面モードからウィンドウモードに切り替えると、HDX 3D Proセッションが数分間応答しなくなることがありました。

[#LC3193]

- インストールしてないにもかかわらず、HDX 3D Proが予期せずVDA上で起動することがありました。この問題は、Windowsディスプレイドライバーモデル (WDDM) の代わりにWindows XPディスプレイドライバーモデル (XPDM) がVDA上にインストールされていると発生します。

[#LC3215]

#### HDX MediaStream Flashリダイレクト

- Microsoft Internet Explorer 9またはそれ以降で、Flashリダイレクト機能を使用してビデオを再生しているときにほかのビデオのリンクをクリックすると、2つのビデオの音声重複して再生されるという問題がありました。

この修正によりこの問題が解決されることはありませんが、サーバー上で以下のレジストリキーを設定することでこの問題の発生時間を短くすることができます。

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer

値の名前: UrlListForTerminationFlashInst

種類: REG\_MULTI\_SZ

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer名前:

UrlListForTerminationFlashInst種類: REG\_MULTI\_SZデータ: <影響を受けるWebサイトのURL。個別の行ごとに http://と https://の両方を含む。例: http://www.youtube.com/; https://www.youtube.com/>

[#LC0453]

- Microsoft Internet Explorer 9またはそれ以降で、Webページに埋め込まれたビデオでのFlashリダイレクトに失敗することがありました。この問題が発生すると、再生画面が黒く表示されます。

この修正を有効にするには、以下のレジストリキーを設定する必要があります。

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer

値の名前: WorkWithAeroEnabled

種類: REG\_DWORD

値のデータ: 0

[#LC0477]

- Microsoft Internet Explorer 8およびそれ以降でYouTubeの検索ボックスを使用すると、Flashリダイレクト機能がサーバー側での処理に切り替わるという問題がありました。

この修正を有効にするには、サーバー上で次のレジストリキーを設定します。

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer

値の名前 : SupportedUrlHeads

種類 : REG\_MULTI\_SZ

データ : <個別の行での値ごと、null区切り>

http://

https://

file://

注 : SupportedUrlHeadsは参照番号#LA4151の修正で最初に追加されました。このレジストリ値はこの修正でも使用されています。レジストリにSupportedUrlHeadsを追加すると、このレジストリ値で指定されたURLを除くFlashのインスタンス化要求がFlashリダイレクト機能で無視されるようになります。

[#LC0505]

- cnn.comやmsn.comに埋め込まれたビデオでFlashリダイレクト機能が正しく動作しないという問題がありました。この問題が発生すると、再生画面が黒く表示されます。

この問題を解決するには、この修正の適用後、以下のレジストリキーを設定します。

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ASJSCallbacks

cnn.comの場合 :

名前 : cnn.com

種類 : REG\_MULTI\_SZ

データ :	javaScriptLoaded
	setQueueAutoplay
	setAdVisibility
	setPlayerMode
	setAdSection
	setAdKeyValue
	playContent
	getContentEntry
	switchTrackingContext

msn.comの場合 :

名前 : msn.com  
種類 : REG\_MULTI\_SZ

データ : MsnVideoCallback  
  
MsnVideoPropertyCallback  
  
GetWidgetID

注 : この修正は、Internet Explorer 9以降でのみ有効です。

[#LC0665]

- Microsoft Internet Explorer 10で、cnn.com Webページに埋め込まれたビデオでのFlashリダイレクトに失敗することがありました。これは、Webサイトで最初のビデオが自動的に再生された時に発生します。タイトル、状態、および広告といったこのビデオの一部の内容が見られなくなることがありました。

[#LC0830]

- m.mlb.comやmsn.comに埋め込まれたビデオでFlashリダイレクト機能が正しく動作しないという問題がありました。この問題が発生すると、再生画面が黒く表示されます。

この修正を有効にするには、次のレジストリキーを設定します。

mlb.comの場合 :

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ASJSCallbacks  
名前 : m.mlb.com  
種類 : REG\_MULTI\_SZ  
データ : <個別の行での値ごと、null区切り>  
setPlaylist  
startPlaylist

msn.comの場合 :

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ASJSCallbacks  
名前 : msn.com  
種類 : REG\_MULTI\_SZ  
データ : <個別の行での値ごと、null区切り>  
MsnVideoCallback  
MsnVideoPropertyCallback  
GetWidgetID

この修正は、Internet Explorer 9およびそれ以降でのみ有効です。これを実行できるようにするには、Internet Explorerをエミュレーションモードで実行するように設定し、Internet Explorer 9と10をドキュメントモードに設定します。詳しくは、Citrixサポートサイトの[CTX136588](https://support.citrix.com/article/CTX136588)を参照してください。

[#LC1043]

- 次のようなWebサイトでビデオを再生しようとすると、Internet Explorerが予期せず閉じることがありました。



- 「アクセラレーションを無効にする」オプションが有効になっている、またはサイトがブラックリスト化されている。
- ユーザーが、FlashリダイレクトをサポートしていないReceiverのバージョン (Receivers for Mac OS X、iPad、iPhone、およびAndroidなど) を使用している。

[#LC2256]

## HDX MediaStream Windows Mediaリダイレクト

- このリリースは、HDXグラフィックエンコーダーにより生成された冗長なフレームを修正し、フレームレートをよりスムーズなものにしてインタラクティブ性を向上させます。

[#LC2177]

- Receiver for Windowsのバージョン13.3または13.4によるVDAセッションでビデオファイルを再生すると、カスタムアプリケーション内のオーディオとビデオ間で同期が起こります。ただし、Receiverをより新しいバージョンにアップグレードすると、Windows Mediaリダイレクトに失敗します。この問題が発生した後は、ユーザーが再生できるのはオーディオのみまたはビデオのみとなります。

この修正を有効にするには、Citrix Receiver for Windows 4.3を使用する必要があります。

[#LC2516]

- ビデオ再生が一時停止状態のままWindows Media Playerを終了すると、Windows Media Playerが応答不能になることがあります。

[#LC0728、#LC3410]

## インストール、アンインストール、アップグレード

- VDA for Desktop OSに対する従来のアップデートでは、WDDMドライバーのインストールが試行されました。WDDMドライバーのインストールを無効にする/nocitrixwddmオプションを指定してインストールしたVDAにこれらのアップデートを使用すると、インストーラーがループ状態に陥り、VDAが繰り返し再起動されるという問題がありました。この修正により、/nocitrixwddmオプションを指定してインストールしたVDAをアップデートするときにWDDMドライバーのインストールが試行されなくなり、この問題が解決されます。

[#LC0443]

- ICAWS750WX64012よりも前にリリースされたVDA Hotfixをインストールすると、Citrix Remote Multi-Touch DeviceおよびCitrix Multi-Touch Redirection Servicesのサービスの状態がデフォルトに再設定され、手作業で有効または無効にした設定が元に戻ってしまいます。

[#LC0446]

- 2014年9月より前にリリースされたHotfixをXenApp 7.5、XenDesktop 7.1および7.5のVDA Core Services for Windows DesktopやVDA Core Services for Windows Server OSにインストールすると、[ICAセッション] パフォーマンスモニターカウンターが削除されるという問題がありました。この問題により、このパフォーマンスモニターカウンターを使用するツールやプロセスが正しく動作しなくなります。この修正により、以前のHotfixのインストールにより削除されたカウンターが復元されます。また、今後リリースされるHotfixをインストールしてもパフォーマンスモニターカウンターが削除されなくなります。

[#LC0771]

- VDAをバージョン7.6から5.6.400にダウングレードした後、VDAが登録に失敗することがあります。

[#LC1465]

- VDAのアップグレード後、以下が発生します。

- レジストリエントリのEnableReadImageFileExecOptionsExclusionListが「outlook.exe」に変わります。

- Outlookはスマートカードとは連携できません。

この修正により、レジストリEnableReadImageFileExecOptionsExclusionListから変わった「Outlook.exe」は削除されます。このレジストリエントリのその他の文字列はそのまま残ります。

[#LC1655]

- ユーザーがVDAをバージョン5.6.400または5.6.500をアップグレードして、マップされたクライアントドライブからファイルをVDAにコピーし、あるいはドライブから直接ファイルを開くと、次のエラーメッセージが表示されます。

「ハンドルが無効です。」

[#LC1910]

- Servicesスナップインを使ってPrint Manager Serviceを再開しようとするとう失敗します。「このサービスを停止できなかった」ことを示すエラーメッセージが表示され、Print Manager Serviceの状態は「停止中」と表示されます。

[#LC2122]

- XenDesktop 7.5からXenDesktop 7.6にVDAをアップグレードした後、ユーザーが接続しようとするとうエラー1030が表示されます。システムイベントログに、次のメッセージが表示されます。「Citrix ICA Transport Driverがポート2598で無効な転送パケットを受信しました。」

[#LC2501]

- VDA for Desktop OSに対する従来のアップデートでは、WDDMドライバーのインストールが試行されました。WDDMドライバーのインストールを無効にする/nocitrixwddmオプションを指定してインストールしたVDAにこれらのアップデートを適用すると、インストーラーがループ状態に陥り、VDAが繰り返し再起動されるという問題がありました。この修正により、/nocitrixwddmオプションを指定してインストールしたVDAをアップデートするときにWDDMドライバーのインストールが試行されなくなり、この問題が解決されます。

[#LC3052]

- まれに、MSP Hotfixパッケージをインストールまたは再インストールするとCitrix Display Driver (Citrix Systems – WDDM)に削除対処としてマークがつくことがあります。これによりインストールには失敗し、VDAのディスプレイ解像度は使用できないままとなります。

[#LC3312]

## キーボード/マウス

- iPadデバイスでOpenOfficeアプリケーションを使用すると、キーボードの自動表示が機能しないことがありました。

[#LC2717]

## ローカルアプリケーションアクセス

- 全画面モードのデスクトップセッションでは、ローカルアプリケーションアクセスが有効で [ドラッグ中にウィンドウの内容を表示する] 設定が無効な場合、Internet Explorerで写真の上にアプリケーションのアイコンを移動すると高解像度の写真が正しく表示されなくなります。

[#LC1722]

## ログオン/認証

- スマートカード認証によりStoreFrontを介してアプリケーションを開始しようとする時、エラーメッセージ「有効な証明書が見つかりませんでした」が表示されました。これにより、ユーザーはログオンできませんでした。

[#LC1704]

## 印刷

- Citrix Print Manager Service (CpSvc.exe) プロセスが予期せず終了することがありました。

[#LA5682、#LC1770]

- ネットワークプリンターを追加することなく設定される「セッションプリンター」のポリシーは、ほかの「セッションプリンター」のポリシーで指定されたセッションプリンターの作成をブロックします。

[#LC0705]

- VDAでデフォルトのプリンターを変更した後、VDAを再起動して、ユーザーがICAセッションからログオフしなかった場合は、変更前のデフォルトのプリンターがVDAに表示されます。

[#LC0898]

- この修正により、製品のバージョン7.xでは使用できなかった「従来のプリンター名」ポリシーを再度有効になります。

[#LC1002]

- Citrix Print Manager Service (CpSvc.exe) プロセスが異常終了することがありました。この問題は、pServerNameの値が正しくない場合に発生します。

[#LC1241]

- ポリシーの [汎用ユニバーサルプリンターを自動作成する] が有効な場合に、Citrix Print Manager Service (CpSvc.exe) プロセスで汎用ユニバーサルプリンターが作成されないという問題がありました。

[#LC1255]

- この機能拡張は、Universal Print ServerおよびUniversal Print Clientのパフォーマンスカウンターを実装します。

[#LC1820]

- Citrixユニバーサルプリンタードライバーを使用すると、最初のページが空白で印刷されることがあります。

[#LC2771]

- 「ユニバーサル印刷EMF処理モード」ポリシーは、「EMFスプールファイルを再処理する」と設定した場合でも、デフォルトの値のままに設定します。

[#LC2521]

- ドキュメントを印刷または別のプリンターを選択しようとする時断続的な遅延が発生し、最長で1分間「プリンターに接続しています」というメッセージが表示されることがありました。

[#LC2701]

- 参照番号#LC0898を適用しセッションローミングを構成した後、クライアントプリンターとセッションプリンターでプリン

ターを作成すると失敗します。

[#LC3490]

## Remote PC

- 同じRemotePC VDAに対する既存のリモートデスクトップとコンソールセッションがある場合にユーザーがReceiverセッションを開始すると、「Reconnecting...」というメッセージが表示されてReceiverセッションに失敗します。このメッセージは、ユーザーが手動でReceiverセッションを閉じるまで表示されます。

[#LC0627]

- 再起動後、ログオン画面が表示されている間のおおよそ30秒以内に物理的なRemote PCにログオンすると、元のセッションの持続期間中はVDAへのRemote PC ICAセッションを確立できません。

[#LC3603]

## シームレスウィンドウ

- Windowsタスクバーなどドックされたアプリケーションを移動すると、最大化されたシームレスアプリケーションが正しくサイズ変更されません。

[#LC1342]

- アプリケーション処理が実行中であってもシームレスウィンドウは表示されず、VDA上でセッションが終了します。

[#LC1728]

## セッション/接続

- Access/NetScaler Gateway経由で接続したセッションにAccess/NetScaler Gatewayを使用せずに再接続するなど、前回接続時と異なる方法でVDAセッションに再接続すると、Citrixポリシーが正しく再評価されないという問題がありました。

注：この修正を有効にするには、WindowsデスクトップOSまたはWindowsサーバーOSデバイスの以下の場所のすべてのエントリを削除する必要があります。説明されている問題が起きていない場合はこの修正を適用しないでください。

- レジストリ：HKEY\_LOCAL\_MACHINE\SOFTWARE\Policies\Citrix
- フォルダ：C:\ProgramData\Citrix\GroupPolicy\\*.\*

[#LA5051]

- 同じ画面解像度が設定されたデュアルモニター構成の2つのエンドポイントで、一方のエンドポイントで切断したセッションに他方のエンドポイントから再接続すると、すべての最上位ウィンドウがプライマリモニター上に移動するという問題がありました。

[#LA5440]

- 公開アプリケーションの実行中にexplorer.exeコマンドを実行してWindowsエクスプローラーを起動すると、エクスプローラーではなくデスクトップが開くという問題がありました。

注：この修正をインストールした後でexplorer.exeプロセスが異常終了することがあります。詳しくは、Knowledge Centerの[CTX128009](#)を参照してください。

[#LC0285]

- Receiverセッションに再接続した場合、多くのWindows ExplorerインスタンスをVDA上で実行しているとCPU消費量が高く

なることがあります。さらに、再度Windowsのタスクバー内に最小化する前に、最小化されたウィンドウが断続的に最大化されます。

[#LC0315]

- 接続したUSB磁気カードリーダーが仮想デスクトップ上で認識されたときに、正しいドライバーがロードされないという問題がありました。

[#LC0360]

- Windowsキーと方向キーを同時に押してアプリケーションのウィンドウを画面の隅に移動してからセッションを切断し、そのセッションに再接続すると、アプリケーションウィンドウのサイズが変更されるという問題がありました。この問題では、サイズ変更後のウィンドウサイズはアプリケーションにより異なり、サイズの差は非プライマリモニター上で顕著です。

[#LC0386]

- ユーザーセッションが応答不能になり、ログオフできなくなることがありました。この問題は、コンソールから手作業でログオフするときにも発生します。

[#SLC0472]

- USBリダイレクトポリシーで [デバイス接続を単純にする] を無効にすると、SafeNet UKey USBデバイスが正しく動作しなくなるという問題がありました。

[#LC0480]

- ユーザーがタスクバーから公開アプリケーションのVisual Basic Editorを閉じると、公開されているExcelのスプレッドシートも閉じました。

[#LC0534]

- SDカードがユーザーデバイスに挿入されVDAセッションが開始したら、カードがクライアントドライブマッピングによりリダイレクトされると、Windows Explorerではフロッピーディスクとして表示されます。

この修正を有効にするには、以下のレジストリキーを設定します。

- 32ビットシステムの場合：  
HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\UNCLinks  
値の名前：DisableFLoppyIcon  
種類：REG\_DWORD  
値のデータ：1
- 64ビットシステムの場合：  
HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\UNCLinks  
値の名前：DisableFLoppyIcon  
種類：REG\_DWORD  
値のデータ：1

[#LC0616]

- ユーザーデバイスで [地域と言語]、[日付と時刻の形式] のコンボボックスのコントロールをリモート操作しようとするとき失敗します。この問題は、コンボボックスのコントロールのリモート操作に一度成功して、2度目にリモート操作を試そうとしたときに発生します。

[#LC0836]

- リダイレクトCOMポートデバイスは、Receiverセッション内では動作しないことがありました。

[#LC0851]

- VDAが応答しなくなると、Receiverから「接続が確立されました。機能をネゴシエートしています。」というメッセージを表示するポップアップウィンドウが開きます。そのほかのセッションを正常に開始するには、VDAを再起動します。

[#LC0909]

- ユーザーがReceiver for Windowsからログオンしている場合、タイムゾーンは正しくありません。このHotfixを正常に機能させるため、次のものをインストールする必要があります。
  - ユーザーデバイスとサーバー上に同じMicrosoftタイムゾーン更新Hotfixをインストールします。たとえば、Microsoft Hotfix [KB2998527](#)がユーザーデバイスにインストールされている場合は、サーバーにもこのHotfixをインストールします。
  - サーバーオペレーティングシステムがWindows Server 2008 R2 Service Pack 1の場合は、サーバーにMicrosoft Hotfix [KB2870165](#)をインストールします。
  - ユーザーデバイス上に参照番号#LC1392がインストールされます。

[#LC1061]

- USBリダイレクトポリシーに1000文字を超える規則を指定すると、禁止されているデバイスを含めすべてのUSBドライブがリダイレクトされるという問題がありました。

[#LC1153]

- タスクバー上のExcelアイコンの問題を修正する機能 (Excelhook) が有効になっている場合、Microsoft Project 2013で複数のプロジェクトを開くと、最初に開いたプロジェクトのアイコンがタスクバー上に表示されなくなることがありました。

[#LC1158]

- WAN接続でデスクトップコンポジションリダイレクト (DCR) を使用すると、文字とグラフィックを含んでいるWebページやドキュメントの表示画面が応答不能になり、帯域幅が過度に生じられることがありました。

[#LC1256]

- Desktop Delivery Controller (DDC) で、ポリシーの [表示品質] 設定で [操作時は低品質] を選択し、 [視覚的無損失の圧縮を使用する] 設定を有効にすると、最終イメージの品質が低下することがありました。

[#LC1271]

- ユーザーのReceiverセッションへの初回ログオン時に画面の表示が乱れることがありました。この問題は、 [表示品質] 設定で [操作時は低品質] を選択し、 [視覚的無損失の圧縮を使用する] 設定を有効にすると発生します。

[#LC1272]

- サーバーVDAセッションで2モニターセットアップが使用されている場合、セカンダリモニターを接続または取り外すとプライマリモニターが応答しなくなることがありました。

[#LC1359]

- ユーザーが切断されたセッションに接続しようとする、接続の確立に3分間かかることがあります。

[#LC1376]

- ユーザーはリモートデスクトップ (RDP) セッションからReceiverセッションには切り替えることができません。

[#LC1383]

- XenDesktopおよびXenAppセッションのMailTo機能について、リダイレクトが正常に機能しないことがあります。ユーザーがWebブラウザで [MailTo] リンクをクリックすると、公開されたMicrosoft Outlookが開く代わりにエラーが表示されま

[#LC1400]

- 公開アプリケーションを起動すると、Citrix Graphics (CtxGfx.exe) プロセスによりCPU消費量が通常よりも高くなります。

[#LC1404]

- 残留セッション機能が有効でアプリケーションがiOSまたはAndroidデバイスで閉じられると、デバイスの画面に黒いウィンドウが表示されたままとなります。

[#LC1407]

- ユーザーデバイス上のUSBストレージデバイスがXenDesktop 5.6に接続されている場合、"UNCEnabled"のレジストリ値が"0"に設定され、"InitialClientDriveLetter"のレジストリ値が"N"に設定されていると、セッションのUSBディスクのドライブ文字は"N"の代わりにデフォルトの"V"から始まります。

詳しくは、Knowledge Centerの[CTX127968](#)および[CTX122061](#)を参照してください。

[#LC1451]

- ユーザーがデスクトップセッションからログオフすると、WFSHellWindowが応答を停止してメッセージ「Wfshell needs to close」が3分間表示されます。

[#LC1591]

- VDAがスクリーンセーバーまたはパワーセーブモードになった後、ブランク画面のセッションを更新する入力 (マウスまたはキーボード) を実行するまでは、画面上に情報が表示されたままとなります。これは、DWORD値の HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\Graphics\SetDisplayRequiredMode = 0でセッションのスクリーンセーバーおよびパワーセーブモードオプションが有効になっている場合に発生します。

[#LC1650]

- マップされたドングルにアクセスすると、サードパーティのアプリケーションがXenDesktopセッションで応答しなくなることがありました。

[#LC1657]

- レジストリでUxPersistenceキーが有効になっている場合、セッションを切断してから再接続した後は、Windowsキーおよび左または右矢印キーを使ってアプリケーションを画面の端に移動しようとする失敗します。

[#LC1684]

- VDIに対してReceiver接続からリモートデスクトップ (RDP) セッションに切り替える場合、RDP接続の転送ログオンダイアログボックスでユーザーが [いいえ] を選択すると、Receiverへの再接続は状態1030で失敗します。

[#LC1696]

- ユーザーがXenDesktopからログオフした時、XenCenterまたはvSphereコンソールにユーザーデスクトップセッションが数秒

表示されます。

[#LC1782]

- スマートカードを使ってユーザーがログオンしてからログオフし、次にユーザー名とパスワードを使ってログオンすると、接続が開始してかほばすぐに切断します。この問題は、ユーザーがスマートカードを使ってもう一度接続するまで続きません。

[#LC1831]

- ユーザーデバイスのキーボードレイアウトがVDAのキーボードレイアウトとは異なる場合、Receiver for Windowsから仮想デスクトップセッションを開始する間、約60~70秒の間表示が応答しなくなることがありました。

[#LC1939]

- WMI Provider ServiceのCPU消費量がWindowsサーバーやデスクトップVDよりも高くなり、これによりVDAが応答しなくなります。また、Receiver接続に失敗します。

[#LC2041]

- Citrix Receiver for LinuxまたはCitrix Receiver for Windows CEを使用する画面解像度がより小さいデバイスでCitrix Receiver for Windowsを使用するセッションからユーザーが切断し、同じセッションに再接続すると、セッション画面の解像度のサイズ変更で失敗することがありました。この問題は、「従来のグラフィックモード」ポリシー設定がセッションに対して有効な場合に発生します。

[#LC2079]

- Lotus NotesをReceiver for iOSがインストールされたiPadで起動すると、このアプリケーションが予期せず閉じることがありました。

[#LC2124]

- XenDesktopセッションで、サービスが示すメッセージを参照するかどうかわからないメッセージボックスが表示されます。[メッセージを表示]をクリックすると、XenDesktopセッションで黒い画面になることがありました。これが起きるのは、XenDesktop VDAがすでにセッション0を使用している場合です。

[#LC2160]

- 従来のグラフィックが有効な場合、マウスポインタのシャドウは、自動で無効になりません。

[#LC2222]

- ユーザーがセッションから切断すると、Citrix WinFrame Shell (WFShell.exe) によるCPU使用量が増えることがありました。

[#LC2235]

- Web InterfaceにVDAデスクトップセッション内からログオンして、同じデスクトップを開こうとすると、セッションが切断し、VDAが応答不能になることがありました。そのためデスクトップを起動できず、VDAを再起動する必要がありました。

[#LC2267]

- 入力言語のデフォルトのUnicodeがユーザーデバイスとVDAで異なる場合、セッションデスクトップがロードしている間、セッションが応答不能になることがありました。



[#LC2287]

- クライアントドライブマッピング (CDM) から公開アプリケーションを開き、次に文字列を入力後にファイルを保存すると、変更時間のスタンプは更新されません。

[#LC2305]

- Wfshell.exeは、ユーザーがセッションから切断後も引き続き大容量のCPUリソースを使用します。

[#LC2391]

- USBリダイレクトを有効にして、Wacomタブレットとペンを使おうとすると失敗します。

[#LC2409]

- キーボードのレイアウトがユーザーデバイスとVDAで異なる場合、セッションデスクトップがロードしている間、セッションが応答不能になることがありました。

[#LC2466]

- リモートデスクトップ (RDP) 8を有効にしており、ネットワークレベル認証 (NLA) を使用していない場合、Receiverから切断した後、RDPで接続に失敗するという問題がありました。

[#LC2480]

- ICAプロキシが有効な場合に、シングルサインオンを使用してNetScaler GatewayからStoreFrontに接続し、ログオフしてから再度ログオンすると、VDAポリシーが機能しないという問題がありました。切断後に再接続するとポリシーが動作しません。

[#LC2577]

- タスクバー上のExcelアイコンの問題を修正する機能が有効な場合、公開されている複数のExcelワークブック間で切り替えると、選択したワークブックのフォーカスが失われることがありました。さらに、公開されているExcelワークブックをどれも選択できず、フォーカスは最後に選択されたファイルにとどまります。

[#LC2589]

- 暗号レジストリキーを更新後、ユーザーはアプリケーションを開始できません。この修正を適用するには、SSL SDKの新しいバージョンをインストールしてください。

[#LC2729]

- VDAに再接続する際、[User Experience Persistence (ユーザーエクスペリエンス保持)] が有効になっている場合でも、アプリケーションウィンドウの状態が維持されず、たとえばウィンドウがそのままの大きさではなく、最大化されてしまうことがありました。

[#LC2769]

- COMポートを経由してPOSデバイスが共有デスクトップにリダイレクトされると、共有デバイス上で実行中のサードパーティソフトウェアからデバイスが認識されないことがありました。

[#LC2775]

- Windows Phone 8.1でInternet Explorerの公開されたバージョンを開くと、デバイスが横向きの場合、画面の半分が空白のウィンドウとして表示されるという問題がありました。Internet Explorerウィンドウを最大化することもできません。デバ

イスを縦向きにすると、画面は正常に表示されます。

[#LC2786]

- Microsoft OfficeアプリケーションをVDA for Windows-based Desktop OSで実行している場合に予期せず閉じることがありました。

[#LC2878]

- ラップトップで外部モニターを使用している場合、外部モニター画面の隅に白い矩形が表示されることがあります。この問題は、とりわけ外部モニターの画面解像度がプライマリモニターの解像度と違う場合に起こります。矩形が表示されるのはプライマリモニターの左側と、プライマリモニターの上端です。

[#LC3186]

- Citrix APIフックとの相互通信のため、Microsoft Officeアプリケーションが時折予期せずに実行を停止することがありました。

[#LC3280]

- USBのリダイレクトを無効にして、公開アプリケーションから切断すると次のエラーが発生し、イベントビューアーに偽のイベント261が表示されます。

「Citrix Device Redirector ServiceはRedirector BusでIOオペレーションを完了できませんでした。」

[#LC3439]

#### サイト/ファームの管理

- すべての権限を持つカスタム管理者はファームからオフラインサーバーを削除できず、PowerShellコマンドの"Remove-XAServer"の実行時にアクセス拒否エラーメッセージを受信します。

- [#LC0752]

- サーバーに読み取り専用のドメインコントローラーへのアクセス権限しかない場合、ポリシーの処理に時間がかかることがありました。

[#LC0828]

- RESソフトウェアを使って展開を管理し、Instant LogOffポリシーを有効にしている場合、ユーザーのセッションログオフ時にXenDesktop Controller上のCitrix Broker Serviceセッション情報を更新できず、Controllerに不正確なセッション状況が反映されることがありました。

[#LC1173]

- このHotfixを適用すると、ユーザーデバイスの永続レジストリ設定を作成できるようになります。

[#LC1827]

- 低速接続の場合、グループポリシーエンジン (CitrixCseEngine.exe) によりログオン時に遅延が生じることがありました。

この修正を有効にするには、次のレジストリキーを作成します。

- HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\GroupPolicy  
値の名前: GpoCacheEnabled

種類：REG\_DWORD

値のデータ：1

- HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\GroupPolicy

値の名前：CacheGpoExpireInHours

種類：REG\_DWORD

値のデータ：5 ~ 24

[#LC1987]

## スマートカード

- VDAのアップグレード後、スマートカードを使用してMicrosoft Outlookにログオンしようとする時、暗証番号 (PIN) の入力が求められません。この場合、次のようなメッセージが表示されます。

「スマートカードを挿入してください。」

[#LC0556]

- スマートカードリーダーにカードを挿入した状態でCitrix Receiver for Linuxのエンドポイントからログオンすると、セッションが応答不能になることがありました。

[#LC0982]

- scardhook64.dllで発生する例外により、ActiVidentityプロセス (accompks.exe) が応答不能になることがありました。この問題が発生すると、スマートカードで認証するWebサイトを使用できなくなり、VDAの再起動が必要になります。

[#LC1056]

- この機能拡張により、エンドポイントデバイスからのスマートカードリーダーの取り外しが、スマートカードの取り出しと同等に処理されるようになります。これにより、VDAのポリシーでスマートカード取り出し時の動作として [ワークステーションのロック] を指定すると、スマートカードリーダーの取り外しによりセッションが切断され、VDAがロックされるようになります。

この修正を有効にするには、以下のレジストリキーを設定します。

- HKEY\_LOCAL\_MACHINE\Software\Citrix\SmartCard\

値の名前：ReaderUnpluggedDetectEnable

種類：REG\_DWORD

値：prop 2 value

- HKEY\_LOCAL\_MACHINE\Software\Citrix\SmartCard\

値の名前：ReaderUnpluggedDetectPollingInterval

種類：REG\_DWORD

デフォルトのレポート間隔は5秒です。

[#LC1059]

- この機能拡張により、参照番号#LA1059の修正で追加された機能が拡張され、Windows 8.1のエンドポイントからVDA Version 7.5への接続に対しても同様の修正が適用されるようになります。

#LA1059の内容は以下のとおりです。

この機能拡張により、エンドポイントデバイスからのスマートカードリーダーの取り外しが、スマートカードの取り出しと同等に処理されるようになります。これにより、VDAのポリシーでスマートカード取り出し時の動作として [ワークステーションのロック] を指定すると、スマートカードリーダーの取り外しによりセッションが切断され、VDAがロックされるようになります。

この修正を有効にするには、以下のレジストリキーを設定します。

- HKEY\_LOCAL\_MACHINE\Software\Citrix\SmartCard\  
値の名前：ReaderUnpluggedDetectEnable  
種類：REG\_DWORD  
値：prop 2 value

- HKEY\_LOCAL\_MACHINE\Software\Citrix\SmartCard\  
値の名前：ReaderUnpluggedDetectPollingInterval  
種類：REG\_DWORD  
デフォルトのレポート間隔は5秒です。

[#LC1713]

- VDAサーバーのicardd.dllでブルースクリーンエラーが発生することがありました。

[#LC2277]

- 暗証番号 (PIN) 付きのスマートカードを使用したVDAセッションへの再接続が失敗し、以下のいずれかのエラーメッセージが表示されます。

- スマートカードを挿入してください
- カードを読み込んでいます

ユーザー名とパスワードを使用してログオンすると、セッションが接続されます。

[#LC2596]

#### システムの例外

- サーバーのpicadm.sysで、バグチェックコード0x22によるブルースクリーンエラーが発生することがありました。

[#LC0227]

- パススルーセッションのMedia Playerでビデオを再生すると、セッションが予期せずに終了する場合があります。

[#LC0553]

- 公開アプリケーションのMicrosoft Excel 2010でスプレッドシート間を移動するとエラーが発生することがありました。

[#LC0809]

- VDAのctxdvc.sysでブルースクリーンエラー (バグチェックコードE3) が発生することがありました。

[#LC0857]

- グラフィックアプリケーションを使用するときにVDAセッションが最大化されると、VDAが応答不能になることがありました。

[#LC0908]

- 以下のいずれかの場合、Windowsエクスプローラーは予期せずに閉じてしまうことがあります。

- 名前に260文字を越える文字が含まれているファイルを大量に選択して、[送る] の [Fax受信者] オプションを選択した場合。
- サードパーティのアプリケーションを開こうとした場合。
- Nitro PDFを使用して、ファイルを結合しようとした場合。

[#LC0938]

- ICA WSK Transport Driver (tdwsk.sys) のページフォールトにより、VDAでブルースクリーンエラーが発生します。

[#LC1032]

- VDAをバージョン7.6から5.6.400にダウングレードした後、オペレーティングシステムでvd3dk.sysエラーが発生し、停止コード0x3Bと共にブルースクリーンが表示されます。

[#LC1200]

- オペレーティングシステムでpicadms.sysエラーが発生し、ブルースクリーンが表示されます。この問題は、TrucationResourceがnullになる場合に発生します。

[#LC1430]

- ICAセッションでSSLが有効になっている場合、Receiver for HTML5からオーディオファイルを再生するとVDAがブルースクリーンのエラーで失敗することがありました。

[#LC1752]

- まれに、VDAが予期せずシャットダウンして、再起動することがあります。

{#LC1876}

- オペレーティングシステムのctxdvc.sysで、バグチェックコード3Bによるブルースクリーンエラーが発生することがあります。

[#LC2119]

- オペレーティングシステムのtdica.sysで、バグチェックコード44[MULTIPLE\_IRP\_COMPLETE\_REQUESTS]によるブルースクリーンエラーが発生することがあります。

[#LC2171]

- Windows Server 2008 R2およびWindows Server 2012 R2では、Receiver for Windowsで接続すると重大なエラーが発生することがありました。

[#LC2179]

- VDAをバージョン5.6 Rollup 4から7.6にアップグレード後、シャットダウン中にVDAが応答しなくなることがあります。

[#LC2245]

- サーバーのicausb.sysでエラーが発生し、ブルースクリーンにメッセージPNP\_DETECTED\_FATAL\_ERRORが表示されることがありました。

[#LC2811]

- wfshell.exeプロセスが異常終了し、新規の接続ができなくなることがありました。

[#LC3024]

- 切断されたセッションに再接続すると、グループポリシーエンジン (CitrixCseEngine.exe) が予期せず終了することがありました。

[#LC3091]

- グラフィックアプリケーションを使用するときにVDAセッションが最大化されると、VDAが応答不能になることがありました。

[#LC3279]

- VDAのvdtw30.dllでブルースクリーンエラーが発生することがありました。

[#LC3354]

- VDAでvd3dk.sys上に致命的な例外が発生し、停止コード0x50と共にブルースクリーンが表示されることがありました。この問題は、スタートアップ時にJavaベースのアップグレードツールがVDAをアップグレードするために構成されている場合に発生します。

[#LC3372]

- セッションへ接続しようとする、explorer.exeプロセスが予期せずに終了する場合があります。

[#LC3411]

## ユーザーエクスペリエンス

- この修正により、Receiver for Windowsのタスクバーでのアプリケーションの表示に関する以下の問題が解決されます。

- アプリケーション固有のアイコンではなくReceiverのデフォルトアイコンが表示される。
- 同一セッション内で複数のアプリケーションを実行する場合にReceiverの単一のデフォルトアイコンにグループ化（結合）される。

[#LC0094]

- LAN接続でReceiver for Windows Version 4.200よりも前のバージョンを使用すると、デスクトップコンポジションリダイレクト機能で操作上の問題が生じることがありました。

[#LC1507]

## ユーザーインターフェイス

- VDAへの再接続後や、複数のモニターを使用している場合、ウィンドウの配置が維持されないことがあります。

[#LC2075]

- Receiver for WindowsまたはモバイルReceiverから公開アプリケーションを開くと「キーボードの自動表示」が表示されないことがありました。

[#LC2747]

## その他

- この修正では、vdaredirector.exeの実行時にメッセージウィンドウを表示させるvdaredirector.exeに対する/Verboseオプションが導入されています。/Verboseオプションを指定するまで、メッセージウィンドウは表示されません。

[#LC0338]

- Microsoft Windows 8 (32ビットバージョン) VDAにサードパーティのアンチウイルスソフトウェアをインストールした後、Citrix Receiverセッションを開始できません。

[#LC1636]

- 参照番号#LA5736に対する修正を含むHotfixまたは以下のいずれかのHotfixがVDAにインストールされている場合、コンソールセッションでLyncステータスを変更できません。

- ICAWS750WX64007
- ICAWS750WX64008
- ICAWS750WX64012
- XD710ICAWSWX64003

Lyncステータスは常に「退席中」と表示されます。

重要：このHotfixの適用後、展開にNetScaler Gatewayが含まれる場合は、ユーザーがVDAセッションを切断してコンソールセッションに接続すると、Lyncステータスは引き続き「退席中」と表示されます。

[#LC1691]

- 参照番号#LA5736に対する修正を含むHotfixまたは以下のいずれかのHotfixがVDAにインストールされている場合、コンソールセッションでLyncステータスを変更できません。

- ICAWS750WX64007
- ICAWS750WX64008
- ICAWS750WX64012
- ICAWS750WX64028
- XD710ICAWSWX64003

Lyncステータスは常に「退席中」と表示されます。

[#LC2479]

- このリリースには、新しいICA仮想チャネルであるFramehawk仮想チャネルのサポートが含まれます。Framehawk仮想チャネルはCitrix HDXテクノロジーを拡張し、パケット損失と遅延の問題がよく発生するブロードバンドワイヤレス接続でユーザーエクスペリエンスを向上させます。詳しくは、「[新機能](#)」を参照してください。

[#LC2782]

- この機能拡張により、一部の5-in-1型のCOMポートデバイスがサポートされます。これにより、VDAセッションでこのデバイスのIDカードリーダー機能を使用できるようになります。

[#LC0325]

- 可視性に制限のある入れ子状態のグループに間接的に属している場合、VMホストアプリを起動しようとしても失敗します。

[#LC3589]

XenAppおよびXenDesktopの7.6 Feature Pack 3 (7.6.300) と7.6 LTSRに含まれているVDA for Desktop OSは同じものです。XenAppおよびXenDesktop 7.7では、バージョン7.6.300と同じ修正が行われており、さらに互換性が向上しています。

[Broker Agent](#)

[HDX 3d Pro](#)

[シームレスウィンドウ](#)

HDX MediaStream Flashリダイレクト	セッション/接続
HDX MediaStream Windows Mediaリダイレクト	サイト/ファームの管理
インストール、アンインストール、アップグレード	スマートカード
キーボード/マウス	システムの例外
ローカルアプリケーションアクセス	ユーザーエクスペリエンス
ログオン/認証	ユーザーインターフェイス
印刷	その他

## Broker Agent

- VDAの登録時に初期化中の状態で停止することがありました。この問題は、Citrix Desktop Serviceが数日間継続して実行されている場合に発生します。

[#LC0570]

- 「CName」機能を有効にすると、VDAの登録に時間がかかることがありました。登録時間を短縮するには、この修正を適用してから「CNameSuppressOrgFqdnLookup」値を有効にしてください。重要：「CNameSuppressOrgFqdnLookup」を有効にする場合は、「CName」機能も有効にする必要があります。  
重要：「CNameSuppressOrgFqdnLookup」を有効にした場合、「CName」も有効にする必要があります。

この修正を有効にするには、以下のレジストリキーを作成する必要があります。

- HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\VirtualDesktopAgent  
値の名前：UseCnameLookup  
種類：REG\_DWORD  
値：有効= 1 (無効= 0)
- HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\VirtualDesktopAgent  
値の名前：CNameSuppressOrgFqdnLookup  
種類：REG\_DWORD  
値：有効= 1 (無効= 0)

[#LC0791]

- マルチホームサーバーでの公開アプリケーションの起動が、以下のエラーで失敗することがあります：

「指定されたアドレスにはCitrix XenAppサーバーがありません。」

[#LC1686]

- バージョン7.6のVDAがXenDesktop 5.6サイトのメンバーの場合、ユーザーが接続し、切断し、その後でもう一度接続しようとすると、再接続に失敗し、VDAが登録解除されます。

[#LC1859]



## HDX 3D Pro

- この修正により、HDXネットワークトラフィックの帯域幅キャップは20メガビット/秒 (Mbps) です。

[#LC2780]

## HDX MediaStream Flashリダイレクト

- Microsoft Internet Explorer 9またはそれ以降で、Flashリダイレクト機能を使用してビデオを再生しているときにほかのビデオのリンクをクリックすると、2つのビデオの音声重複して再生されるという問題がありました。

この修正によりこの問題が解決されることはありませんが、サーバー上で以下のレジストリキーを設定することでこの問題の発生時間を短くすることができます。

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer

値の名前: UrlListForTerminationFlashInst

種類: REG\_MULTI\_SZ

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer名前:

UrlListForTerminationFlashInst種類: REG\_MULTI\_SZデータ: <影響を受けるWebサイトのURL。個別の行ごとに http://と https://の両方を含む。例: http://www.youtube.com/; https://www.youtube.com/>

[#LC0453]

- Microsoft Internet Explorer 8およびそれ以降でYouTubeの検索ボックスを使用すると、Flashリダイレクト機能がサーバー側での処理に切り替わるという問題がありました。

この修正を有効にするには、サーバー上で次のレジストリキーを設定します。

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer

値の名前: SupportedUrlHeads

種類: REG\_MULTI\_SZ

Data: <each value on a separate line, null separated>

http://

https://

file://

注: SupportedUrlHeadsは参照番号#LA4151の修正で最初に追加されました。このレジストリ値はこの修正でも使用されています。レジストリにSupportedUrlHeadsを追加すると、このレジストリ値で指定されたURLを除くFlashのインスタンス化要求がFlashリダイレクト機能で無視されるようになります。

[#LC0505]

- Microsoft Internet Explorer 10で、cnn.com Webページに埋め込まれたビデオでのFlashリダイレクトに失敗することがありました。これは、Webサイトで最初のビデオが自動的に再生された時に発生します。タイトル、状態、および広告といったこのビデオの一部の内容が見られなくなることがありました。

[#LC0830]

- 次のようなWebサイトでビデオを再生しようとする、Internet Explorerが予期せず閉じることがありました。
  - 「アクセラレーションを無効にする」オプションが有効になっている、またはサイトがブラックリスト化されている。
  - ユーザーが、FlashリダイレクトをサポートしていないReceiverのバージョン (Receivers for Mac OS X, iPad, iPhone、およびAndroidなど) を使用している。

[#LC2256]

## HDX MediaStream Windows Mediaリダイレクト

- Receiver for Windowsのバージョン4.1または4.1.2をインストールした後でHDX MediaStream Multimedia Accelerationが有効な場合は、Windows Media Playerはビデオを読み込みますが、最初のフレームより後のビデオは処理できません。

[#LC1460]

- このリリースは、HDXグラフィックエンコーダーにより生成された冗長なフレームを修正し、フレームレートをよりスムーズなものにしてインタラクティブ性を向上させます。

[#LC2177]

- Receiver for Windowsのバージョン13.3または13.4によるVDAセッションでビデオファイルを再生すると、カスタムアプリケーション内のオーディオとビデオ間で同期が起こります。ただし、Receiverをより新しいバージョンにアップグレードすると、Windows Mediaリダイレクトに失敗します。この問題が発生した後は、ユーザーが再生できるのはオーディオのみまたはビデオのみとなります。

この修正を有効にするには、Citrix Receiver for Windows 4.3を使用する必要があります。

[#LC2516]

- ビデオ再生が一時停止状態のままWindows Media Playerを終了すると、Windows Media Playerが応答不能になることがありました。

[#LC3410]

## インストール、アンインストール、アップグレード

- Universal Print Server Clientがインストール済みのシステムにHotfix Rollup Packs 1~4をインストールすると、セッションプリンターの列挙が遅くなるという問題がありました。

[#LA5977]

- 2014年9月より前にリリースされたHotfixをXenApp 7.5、XenDesktop 7.1および7.5のVDA Core Services for Windows DesktopやVDA Core Services for Windows Server OSにインストールすると、[ICAセッション] パフォーマンスモニターカウンターが削除されるという問題がありました。この問題により、このパフォーマンスモニターカウンターを使用するツールやプロセスが正しく動作しなくなります。この修正により、以前のHotfixのインストールにより削除されたカウンターが復元されます。また、今後リリースされるHotfixをインストールしてもパフォーマンスモニターカウンターが削除されなくなります。

[#LC0771]

- Hotfix XA650R04W2K8R2X64008をインストールすると、以前のバージョンのXenAppで正しく動作していた既存のアイコンが破損していると認識されることがありました。

[#LC1081]

- 参照番号#LA5872に対応したHotfixまたはHotfix Rollup PackをインストールしてからReceiver for Windowsでログオンすると、ユーザーの公開アプリケーションが起動されるまでWindowsのログオンセキュリティ通知が表示されたままとなります。

[#LC1215]

- XA650W2K8R2X64R04をインストール後、まれに検出に失敗し、成功するまで複数回検出が試みられることがあります。

[#LC1369]

- Hotfixes XA650W2K8R2X64R03、XA650W2K8R2X64R04、またはXA650W2K8R2X64R05をインストール後、以下の問題が発生することがありました。

- 以下のレジストリエントリにドイツ語のバイナリを使用すると、イベントビューアが失敗します。
  - HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Publishers\{c6d0a271-2e59-40cd-b451-9000a9f104f9}
  - MessageFileName: C:\Program Files (x86)\Citrix\System32\resource\de\RaveEventMessages.Dll
  - ResourceFileName: C:\Program Files (x86)\Citrix\System32\resource\de\RaveEventMessages.Dll
- オペレーティングシステムは英語を使用し、デフォルトのユーザーアカウントにはドイツ語を使用するように設定されていると、RAVEイベントビューアは英語ではなくドイツ語を使用します。

[#LC2130]

- Hotfix Rollup Pack 5をインストールした後、XTEサービスがエラーとなり、ユーザーセッションが停止します。

{#LC2180}

- XenDesktop 7.5からXenDesktop 7.6にVDAをアップグレードした後、ユーザーが接続しようとするエラー1030が表示されます。システムイベントログに、次のメッセージが表示されます。「Citrix ICA Transport Driverがポート2598で無効な転送パケットを受信しました。」

[#LC2501]

- XML BrokerにXA650W2K8R2X64R05をインストールした後、XMLサービスがユーザーを認証するためにかかる時間が長くなる場合があります。負荷の高い読み込み環境において、この時間の増加によりWeb Interfaceへのユーザーセッションがタイムアウトすることがあります。

[#LC2536]

## キーボード/マウス

- Receiver for Linuxのセッションでチェコ語キーボードを使用する場合、一部のキーが正しく処理されないという問題がありました。以下の文字のマッピングが修正されます。

- dがĎとしてマップされる。
- ňがÑとしてマップされる。
- ěがĚとしてマップされる。

注：この修正をインストールすると、Receiver for Linuxセッション内で開くRDPセッションでShiftキーが正しく動作しなくなる場合があります。この問題を回避するには、Microsoft社の[KB2592687](#)で参照される修正プログラムをインストールしてください。または、Citrix Knowledge Centerの[CTX110281](#)を参照してください。

[#LA3968]

- マウスを使ってローカルアプリケーションと公開アプリケーションを切り替えると、キーボード入力ができなくなることはありません。

[#LA5849]

- マウスポインターがデスクトップの境界上で正しく動作しなくなることはありません。

[#LC0383]

- セッションでマウスからの入力が途絶えることがありました。

[#LC0632]

- ユーザーデバイスにキーボードを表示するため「キーボードの自動表示」ポリシーが構成されている場合に、公開アプリケーションにキーボードが自動的に表示されないことがありました。

注：この問題を完全に修正するには、ユーザーデバイスに参照番号#LC3158をインストールします。

[#LC2432]

- iPadデバイスでOpenOfficeアプリケーションを使用すると、キーボードの自動表示が機能しないことがありました。

[#LC2717]

#### ローカルアプリケーションアクセス

- 全画面モードのデスクトップセッションでは、ローカルアプリケーションアクセスが有効で [ドラッグ中にウィンドウの内容を表示する] 設定が無効な場合、Internet Explorerで写真の上にアプリケーションのアイコンを移動すると高解像度の写真が正しく表示されなくなります。

[#LC1722]

#### ログオン/認証

- スマートカード認証によりStoreFrontを介してアプリケーションを開始しようとする時、エラーメッセージ「有効な証明書が見つかりませんでした」が表示されました。これにより、ユーザーはログオンできませんでした。

[#LC1704]

#### 印刷

- Citrix Print Manager Service (CpSvc.exe) プロセスが予期せず終了することがありました。

[#LA4967、#LA5682、#LC0344、#LC0427、#LC0675、#LC0781、#LC1090、#LC1770]

- ユーザーがほかのデバイスに移動したときに、正しいプリンターがデフォルトのセッションプリンターとして設定されないという問題がありました。

[#LC0225]

- Receiver for Linuxでのユーザーセッションで、以下のプリンターが正しくマップされないという問題がありました。

- Epson-LQ-1050
- Epson PLQ-20

この修正を有効にするには、以下のレジストリキーを設定する必要があります。

HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\ICA Client\PrintingSettings

値の名前：AllowPrinterMappingByName

種類：REG\_DWORD

値のデータ：1

[#LC0389]

- 切断された残留セッションに再接続しようとし、その公開アプリケーションが指定のウィンドウサイズの場合、セッション

プリンターがセッションにマップできませんでした。

[#LC0615]

- アクセス違反エラーにより、Citrix Printer Manager Service (Cpsvc.exe) に失敗しました。

[#LC0660]

- 「残留セッション」ポリシーが有効な場合、セッションに再接続しようとする、ユーザーデバイス上のリダイレクトされたプリンターがマップしませんでした。

[#LC0816]

- この修正により、製品のバージョン7.xでは使用できなかった「従来のプリンター名」ポリシーを再度有効になります。

[#LC1002]

- ユーザーが新しいセッションにログオンすると、プリンターの一覧が表示されません。ユーザーが切断されたセッションに再接続しても、プリンターの一覧が表示されません。

[#LC1226]

- ポリシーの [汎用ユニバーサルプリンターを自動作成する] が有効な場合に、Citrix Print Manager Service (CpSvc.exe) プロセスで汎用ユニバーサルプリンターが作成されないという問題がありました。

[#LC1255]

- この機能拡張は、Universal Print ServerおよびUniversal Print Clientのパフォーマンスカウンターを実装します。

[#LC1820]

- Servicesスナップインを使ってPrint Manager Serviceを再開しようとする、失敗します。「このサービスを停止できなかった」ことを示すエラーメッセージが表示され、Print Manager Serviceの状態は「停止中」と表示されます。

[#LC2122]

- 「ユニバーサル印刷EMF処理モード」ポリシーは、「EMFスプールファイルを再処理する」と設定した場合でも、デフォルトの値のままに設定します。

[#LC2521]

- アクセス違反エラーにより、Citrix Printer Manager Service (Cpsvc.exe) が予期せず停止します。

[#LC2624]

- ドキュメントを印刷または別のプリンターを選択しようとする、断続的な遅延が発生し、最長で1分間「プリンターに接続しています」というメッセージが表示されることがありました。

[#LC2701]

- Citrixユニバーサルプリンタードライバーを使用すると、最初のページが空白で印刷されることがあります。

[#LC2771]

- Receiverでログオンすると、ユーザーはDieboldプリンター (Model P361) で印刷できません。

[#LC2908]

- この拡張機能は、ユーザーがログオンすると1つのポート番号につき1つの自動復元プリンターを作成します。

[#LC2943]

#### シームレスウィンドウ

- 公開されたMicrosoft Excelをシームレスウィンドウモードで実行すると、画面がちらつくことがありました。また、ローカルで実行されているアプリケーションにキーボードのフォーカスが移ってしまうこともありました。

この修正を有効にするには、以下のレジストリキーを作成して2つの値を設定してください。

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshell\TWI\XLMAIN

値の名前 : ClassName

種類 : REG\_SZ

値のデータ : 1

Name : 「ShareFile\_Profile」と入力します。

種類 : REG\_DWORD

値のデータ : 0x00000808 (1)

[#LC0401]

- Windowsタスクバーなどドックされたアプリケーションを移動すると、最大化されたシームレスアプリケーションが正しくサイズ変更されません。

[#LC1342]

- 管理者特権がありCitrix Studioで公開されているアプリケーションはシームレスモードでは起動できません。

[#LC1537]

- シームレスセッションでは、シームレスツールバーを2つ目のモニターにドッキングしようとしても、公開ツールバーのシームレスフラグがオンに設定されるとプライマリモニターにドッキングされました。シームレスフラグについては、Citrixのアーティクル[CTX101644](#)を参照してください。

[#LC1599]

- アプリケーション処理が実行中であってもシームレスウィンドウは表示されず、VDA上でセッションが終了します。

[#LC1728]

- マルチモニター環境では、ユーザーがシームレスウィンドウをセカンダリディスプレイに移動してプライマリまたはセカンダリディスプレイの解像度を変更すると、シームレスウィンドウがプライマリディスプレイに移ることがあります。

[#LC1819]

#### セッション/接続

- クライアントドライブマッピングを介してユーザーがファイルをユーザーデバイスに保存しようとする時、マップされたドライブに保存されたファイルにはデータがまったく含まれないことがありました。

[#LA3566]

- マップされたクライアント側ドライブ上にファイルをコピーした後で、そのドライブの空き領域が正しく表示されなくなる

ことがありました。

[#LA4799]

- GPOで複数のWMIフィルターを構成すると、セッションへの再接続に最大60秒のログオン遅延が発生することがありました。

この問題を解決するには、この修正を適用し、次のレジストリキーを使用して再接続時のポリシー計算を無効にします。

HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\Rade

値の名前：DisableGPCalculation

種類：REG\_DWORD

値のデータ：1

[#LA4932]

- 再接続時に [ConnectTime] が再計算されないという問題がありました。このため、EdgeSightのレポートで、一部のセッションの [セッション作成時のサーバーの実行時間] (SCSD) と [セッションサーバー起動時間] (SSSD) の値が過度に大きくなるがありました。

これを行うには、次のレジストリキーを設定します。

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\Dazzle

値の名前：DisableServerStartupEventForReconnect

種類：REG\_DWORD

値のデータ：1

[#LA5141]

- Receiver for Macを使用する場合、Microsoft Outlookでのオーディオファイルの再生に失敗することがありました。

[#LA5657]

- 公開されたExcelの問題を修正する機能 (Excelhook) が有効な場合に、ブックを切り替えたときにExcelが応答不能になることがありました。

[#LA5708]

- XenAppサーバーを再起動した後でライセンスサーバーへの接続に失敗することがありました。

この修正を有効にするには、以下のレジストリキーを作成する必要があります。

- 32ビットWindows :

HKEY\_LOCAL\_MACHINE\Software\Citrix\VirtualDesktopAgent\

値の名前：RA

種類：REG\_DWORD

32ビットWindows : HKEY\_LOCAL\_MACHINE\Software\Citrix\Licensing名前：RA\_retries 種類：REG\_DWORD値：1 (値が5以下を示す場合はデフォルト値が5とみなされます)

- 64ビットWindows :

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\Rade

値の名前：RA

種類：REG\_DWORD

32ビットWindows : HKEY\_LOCAL\_MACHINE\Software\Citrix\Licensing名前：RA\_retries 種類：REG\_DWORD値：1 (値

が5以下を示す場合はデフォルト値が5とみなされます)

[#LA5773]

- ロックされた公開デスクトップセッションへの再接続に失敗して、次のエラーメッセージが表示されることがありました。

「接続エラー：このアプリケーションは、サーバーで同時接続可能なコネクション数の上限に達しました。システム管理者にお問い合わせください。」

この問題が発生すると、既存のセッションとは別のセッションが新しく作成されます。

[#LC0158]

- 切断セッションの再接続に失敗して、既存のセッションではなく新しいセッションが起動することがありました。この問題は、[Restrict Remote Desktop Services users to a single remote session] 設定が有効な場合に発生します。

[#LC0201]

- VDAへのリモートデスクトッププロトコル (RDP) 接続を確立し、RDPセッションを切断すると、イベントログに以下のエラーメッセージ (イベントID 1048) が記録されることがありました。

「Citrix Desktop ServiceはDDCに再登録しています：

「NotificationManager : NotificationServiceThread : Windows Communication Foundationの障害またはブローカーによる拒否 (DDC : bruin.newer.ln) 」」

[#LC0251]

- アプリケーションの最大インスタンス数を1に設定すると、切断セッションのアプリケーションの再起動に失敗して新しいセッションが起動することがありました。この問題は、クライアントの自動再接続機能により再接続用の内部セッション状態に不正な値が設定されるために発生します。

[#LC0259]

- 公開アプリケーションの実行中にexplorer.exeコマンドを実行してWindowsエクスプローラーを起動すると、エクスプローラーではなくデスクトップが開くという問題がありました。

注：この修正をインストールした後でexplorer.exeプロセスが異常終了することがあります。詳しくは、Knowledge Centerの[CTX128009](#)を参照してください。

[#LC0285]

- 1つ目のUNCドライブからのWindows Substituteコマンドを使って2つ目のドライブが作成されている場合、この2つのマップされたドライブのUNCパス間でファイルをコピーすると、ファイルが破損することがあります。

[#LC0288]

- マルチストリームポリシーを有効にすると、サーバーからユーザーデバイス側へのクリップボードデータの転送に時間がかかることがありました。

この問題を解決するには、この修正を適用してから以下のレジストリキーにより大きな値を設定します。

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server]TSEnabled=dword:00000001

値の名前：InitialOutputSpeed

種類：REG\_DWORD

値：10000 – 1000000 (10KB/秒~1000KB/秒、推奨値：300000)



[#LC0390]

- Citrix Knowledge CenterのCTX133198で説明されているExcelアイコンの問題を修正するレジストリキーを設定すると、Microsoft ExcelとMicrosoft VB for Applicationsの両方が動作するセッションで以下の問題が発生することがありました。
  - VB for Applicationsのウィンドウにフォーカスを移してもExcelのメインウィンドウが最前面に表示される。
  - VB for Applicationsのウィンドウを閉じるとExcelのメインウィンドウも閉じる。

[#LC0398]

- ユーザーのログオフ時にセッションが切断状態で停止することがありました。この結果、セッションを閉じたりサーバーを再起動したりすることができなくなります。

[#LC0399]

- XenApp上でMozilla Firefox 28の公開に失敗し、以下のエラーメッセージが表示されることがありました。

「System.NotSupportedException : データがICO形式ではありません」

[#LC0402]

- Windows Server 2008 R2 (サーバーOS) 上のCitrix XenApp 7.6およびXenDesktop 7.6のVDAで、[ようこそ]画面が表示されたときにCore Servicesが応答不能になることがありました。この問題が発生すると、サーバーへのReceiver接続およびリモートデスクトッププロトコル (RDP) 接続の確立に失敗します。

[#LC0405]

- ユーザーセッションが応答不能になり、ログオフできなくなることがありました。この問題は、コンソールから手作業でログオフするときにも発生します。

[#SLC0472]

- 公開デスクトップでは、状態インジケータ進行状況バーが完了せず、ユーザーがアプリケーションにログオンまたは起動したときに特定のログオン状況メッセージが表示されないことがありました。

[#LC0482]

- Microsoft Silverlight Webアプリケーションでの画面切り替え時に、モバイルデバイス (iOSおよびAndroid) 側のキーボードが自動表示されることがありました。

[#LC0499]

- ユーザーがタスクバーから公開アプリケーションのVisual Basic Editorを閉じると、公開されているExcelのワークシートも閉じました。

[#LC0534]

- 公開アプリケーションの起動時のログオン処理に時間がかかることがありました。

[#LC0596]

- XenApp 6.5ポリシーの [モバイルデバイスでの動作] 設定を有効にすると、Receiver for iOSユーザーのログオフ時に以下の内容のメッセージが表示されるという問題がありました。

「ログオフ時に問題が発生しました。ヘルプデスクにお問い合わせください。」

[#LC0628]

- ユーザーが46文字以上のパスワードを使用すると、Windows Server 2008またはWindows Server 2012のVDA上のXenDesktop 7.xに接続できないという問題がありました。

[#LC0689]

- [タイムゾーンリダイレクトを許可する] オプションが有効な場合に、非英語版のWindows 7デバイス上で動作するセッションの公開Javaアプリケーションでタイムゾーン情報が正しく同期されないという問題がありました。この問題は、GetDynamicTimeZoneInformation APIでReceiverセッションにタイムゾーン情報を正しく返すことができないために発生します。

言語パックを使用するユーザーデバイスでは、その言語と同じ言語をシステムロケールに設定してください。

注：この修正は、ロシアのタイムゾーンには無効です。

[#LC0807]

- クライアント側ドライブのマッピングを有効にした場合、公開されたVBアプリケーションのFileStream()ファンクションで書き込まれたファイルの末尾に不正な文字が追加されることがありました。

[#LC0815]

- ユーザーデバイスで [地域と言語]、[日付と時刻の形式] のコンボボックスのコントロールをリモート操作しようとするとき失敗します。この問題は、コンボボックスのコントロールのリモート操作に一度成功して、2度目にリモート操作を試そうとしたときに発生します。

[#LC0836]

- ユーザーが切断されたセッションに接続しようとするとき、接続の確立に3分間かかることがあります。

[#LC0850]

- 公開アプリケーションの起動時に以下のメッセージが表示されることがありました。

「現在、Citrixサーバーでこの公開アプリケーションの起動要求を処理できません。後でもう一度実行してください。問題が解決しない場合は、管理者に問い合わせてください。」

この変更について：アプリケーションの起動時にwfshellプロセスでのアプリケーションプロパティのチェックに時間がかかることがありました。このため、ネットワークの状態によってはチェックが完了する前にタイムアウトが発生し、エラーメッセージが表示されることがあります。

この修正を適用して以下のレジストリキーを設定すると、アプリケーションプロパティチェックのタイムアウト値を最大4分まで変更できるようになります。

HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\ICAST

値の名前：WaitTimeOutValue

種類：REG\_DWORD

次のコマンドを実行して、Up Delay値をミリ秒単位で指定します。デフォルト：12000、サポートされる最大値：240000>

[#LC0860]

- ユーザーがいずれかのバージョンのモバイルデバイス用のReceiverを使ってログオンし、「キーボードの自動表示」ポリシーが有効な公開アプリケーションのLotus Notesを開くと、[宛先]、[CC]、[BCC] フィールドにメールアドレスを

入力する場合にキーボードが表示されませんでした。

[#LC0899]

- VDAが応答しなくなると、Receiverから「接続が確立されました。機能をネゴシエートしています。」というメッセージを表示するポップアップウィンドウが開きます。そのほかのセッションを正常に開始するには、VDAを再起動します。

[#LC0909]

- ユーザーがReceiver for Windowsからログオンしている場合、タイムゾーンは正しくありません。このHotfixを正常に機能させるため、次のものをインストールする必要があります。
  - ユーザーデバイスとサーバー上に同じMicrosoftタイムゾーン更新Hotfixをインストールします。たとえば、Microsoft Hotfix [KB2998527](#)がユーザーデバイスにインストールされている場合は、サーバーにもこのHotfixをインストールします。
  - サーバーオペレーティングシステムがWindows Server 2008 R2 Service Pack 1の場合は、サーバーにMicrosoft Hotfix [KB2870165](#)をインストールします。
  - ユーザーデバイス上に参照番号#LC1392がインストールされます。

[#LC1061]

- サーバー上のアクティブセッションへの再接続に失敗することがありました。アプリケーションの最大インスタンス数を1に設定すると、追加セッションの起動に失敗し、既存のセッションから切断またはログオフしないと再接続したり新しいセッションを起動したりできなくなるという問題がありました。

[#LC1084]

- ファーム内のサーバーがシャットダウン状態のときにドメイン間の信頼関係を変更すると、アプリケーションの列挙に失敗することがありました。この問題では、UPNによるログオンには成功しますが、「<domain>\<username>」形式でのログオンに失敗します。

[#LC1136]

- USBリダイレクトポリシーに1000文字を超える規則を指定すると、禁止されているデバイスを含めすべてのUSBドライブがリダイレクトされるという問題がありました。

[#LC1153]

- タスクバー上のExcelアイコンの問題を修正する機能 (Excelhook) が有効になっている場合、Microsoft Project 2013で複数のプロジェクトを開くと、最初に開いたプロジェクトのアイコンがタスクバー上に表示されなくなることがありました。

[#LC1158]

- まれに、キャプションフォント、スクロールバーのサイズ、およびメニューの高さといったシームレスアプリケーションの視覚設定がユーザーデバイスの設定と正常に同期しないことがありました。

[#LC1169]

- セッション共有に失敗することがありました。

[#LC1184]

- 別のプロセスによってpicadmsysと同じロックが保持されている場合、ユーザーはセッションからログオフできず、セッションは切断された状態のままとなります。

[#LC1243]

- Desktop Delivery Controller (DDC) で、ポリシーの [表示品質] 設定で [操作時は低品質] を選択し、 [視覚的無損失の圧縮を使用する] 設定を有効にすると、最終イメージの品質が低下することがありました。

[#LC1271]

- ユーザーのReceiverセッションへの初回ログオン時に画面の表示が乱れることがありました。この問題は、 [表示品質] 設定で [操作時は低品質] を選択し、 [視覚的無損失の圧縮を使用する] 設定を有効にすると発生します。

[#LC1272]

- 公開されたバージョンのMicrosoft Access 2007では情報をコピーしようとするとき失敗します。情報はOfficeのクリップボードには表示されません。

[#LC1290]

- 公開アプリケーションのMicrosoft Excel 2010でスプレッドシート間を移動するとエラーが発生することがありました。

[#LC1356]

- XenDesktopおよびXenAppセッションのMailTo機能について、リダイレクトが正常に機能しないことがあります。ユーザーがWebブラウザで [MailTo] リンクをクリックすると、公開されたMicrosoft Outlookが開く代わりにエラーが表示されません。

[#LC1400]

- 残留セッション機能が有効でアプリケーションがiOSまたはAndroidデバイスで閉じられると、デバイスの画面に黒いウィンドウが表示されたままとなります。

[#LC14074]

- クライアント自動再接続機能を使ってユーザーがセッションに再接続すると、クライアント側のタイムゾーンはユーザーデバイスにリダイレクトされず、代わりにサーバー側のタイムゾーンがユーザーデバイスに表示されることがありました。

[#LC1426]

- ユーザーがデスクトップセッションに再接続すると、 [セッション作成時のサーバーの実行時間 (SCSD) ] および [セッション開始時のサーバーの持続時間 (SSSD) ] の値はリセットされません。

[#LC1464]

- ドイツ語のWindowsベースデバイスでは、ドイツ語のMultimedia Rave (RaveEventMessages.dll) サービスにより、イベントログサービスを失敗させるsvchost.exeプロセスが作成することがありました。この問題が発生すると、ユーザーはログインできません。

[#LC1543]

- ユーザーが公開アプリケーションを開始すると、Citrix Stack Control Service (SCService64.exe) が応答を停止します。

[#LC1545]

- ユーザーがデスクトップセッションからログオフすると、WFShellWindowが応答を停止してメッセージ「Wfshell needs to close」が3分間表示されます。

[#LC1591]

- マルチモニター環境で、公開シームレスツールバーアプリケーションの背後にアプリケーションが開くことがありました。修正を有効にするため、参照番号#LC1342および参照番号#LC0491のインストールが必要なことがありました。

[#LC1673]

- Windowsベースデバイスにログオンし、XenApp仮想マシン (VM) に接続すると、TWIモジュール (TWI3.dll) によるCPU使用量が増えました。

[#LC1810]

- 2つのデュアルモニターユーザーデバイスセットアップで、HDX機能がある複数のアプリケーションを起動すると、いくつかのアプリケーションが黒いウィンドウとして表示されることがありました。

[#LC1818]

- 従来のグラフィックモードが無効になった状態でXenAppセッションを録画すると、録画ファイルがロールオーバーされないことがありました。

[#LC1850]

- 残留セッションでタイムアウトが発生して、セッションが予期せず終了することになりウィンドウが黒くなります。

[#LC1903]

- ユーザーが切断されたセッションに接続しようとする時、接続の確立に3分間かかることがあります。

[#LC1908]

- Active Directoryサーバーが読みより専用サーバーとして設定されていると、ユーザーはアプリケーションを起動できず、「接続が確立されました。機能をネゴシエートしています。」というメッセージが表示されたままとなります。

[#LC1964]

- セッションで [クライアントの自動再接続] ポリシーが有効になっている場合に、認証プロンプトが表示されない問題がありました。

この修正を有効にするには、以下のレジストリキーを設定します。

- HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\WFShell  
値の名前: SessionReconnectMinTimeInMilliseconds  
種類: REG\_DWORD  
値: 最大セッション長
- HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\WFShell  
値の名前: LockOnReconnect  
種類: REG\_DWORD  
値のデータ: 1

このレジストリキーを設定した後、サーバーを再起動するかコマンド「gpupdate /force」を実行します。

[#LC1990]

- 公開アプリケーションを起動すると、Citrix Graphics (CtxGfx.exe) プロセスによりCPU消費量が通常よりも高くなります。

[#LC2007]

- Citrix Receiver for LinuxまたはCitrix Receiver for Windows CEを使用する画面解像度がより小さいデバイスでCitrix Receiver for Windowsを使用するセッションからユーザーが切断し、同じセッションに再接続すると、セッション画面の解像度のサイズ変更に失敗することがありました。この問題は、「従来のグラフィックモード」ポリシー設定がセッションに対して有効な場合に発生します。

[#LC2079]

- ユーザーがSAP内で開かれた埋め込みのWordファイルの内容をコピーする際、コピー対象を新規のWordファイルにペーストしようとするとう失敗していました。

[#LC2092]

- ユーザーがWeb Interfaceから公開デスクトップを開始すると、セッションが「次の個人用設定を設定しています：Microsoft Windows Media Player」の段階で応答なくなり、デスクトップからオーディオ機能を実行できませんでした。

[#LC2103]

- Lotus NotesをReceiver for iOSがインストールされたiPadで起動すると、このアプリケーションが予期せず閉じることがありました。

[#LC2124]

- セッション属性"AllowReconnect"が"False"に設定されているため、切断されたセッションに再接続しようとするとうまれに失敗することがありました。

[#LC2174]

- ユーザーがセッションから切断すると、Citrix WinFrame Shell (WFShell.exe) によるCPU使用量が増えることがありました。

[#LC2235]

- /appve引数を使ってInternet Explorerを公開アプリケーションとして起動すると、割り当てパッケージの読み込みに失敗します。

[#LC2292]

- Wfshell.exeは、ユーザーがセッションから切断後も引き続き大容量のCPUリソースを使用します。

[#LC2391]

- AutoCADおよびAutoCAD Civil 3dを使っている場合、マウスポインターが砂時計として表示されたままとまります。

[#LC2438]

- クライアントドライブマッピング (CDM) から公開アプリケーションを開き、次に文字列を入力後にファイルを保存すると、変更時間のスタンプは更新されません。

[#LC2305]

- クライアント側ドライブのマッピング (CDM) ドライブにあるファイルを処理しようとするとう、picadmsysでデットロックが発生し、公開デスクトップが応答しなくなることがあります。

[#LC2312]

- Excel.csvファイルに多数のエントリが含まれている場合、ユーザーデバイスからサーバーへのファイルコピーに時間がかかることがありました。

[#LC2366]

- 公開アプリケーションのExcelまたはInternet Explorerを最大化すると、名前ボックスの上部、数式バー、またはメニューバーが白いバーとして表示されます。

ExcelおよびInternet Explorerでこの問題を修正するには、次のレジストリキーを作成します。

Excel

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshe\TWI\XLMAIN

値の名前：ClassName

種類：REG\_SZ

値：XLMAIN

値の名前：Type

種類：REG\_DWORD

値：0x00020000

Internet Explorer

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshe\TWI\IEFrame

値の名前：ClassName

種類：REG\_SZ

値：IEFrame

値の名前：Type

種類：REG\_DWORD

値：0x00020000

この修正を複数のサーバーに適用するには、レジストリフラグを作成します。詳しくはCitrixサポートサイトで入手できる[Seamless\\_Configuration.pdf](#)の「Seamless Per Application Window Registry Flags」を参照してください。

[#LC2504]

- タスクバー上のExcelアイコンの問題を修正する機能が有効な場合、公開されている複数のExcelワークブック間で切り替えると、選択したワークブックのフォーカスが失われることがありました。さらに、公開されているExcelワークブックをどれも選択できず、フォーカスは最後に選択されたファイルにとどまります。

[#LC2589]

- [1ユーザーにつき1インスタンスのみ許可する] 設定または [クライアントの自動再接続] ポリシーが有効な場合「このアプリケーションのインスタンスが既に実行中です。複数インスタンスの実行は許可されていません。システム管理者に連絡してください。」この問題は、2つのフォレスト間でセッション共有に失敗したために発生します。

[#LC2592]

- Windows 7 64ビットのコンピューター上の公開アプリケーションのInternet Explorer 8で再生する場合、アプリケーション

「Unreal Streaming Media Player」でビデオがブランクとなります。

[#LC2650]

- クライアントドライブマッピングを使って1000ファイル以上をコピーすると、パフォーマンスに問題が生じることがありました。

[#LC2702]

- 暗号レジストリキーを更新後、ユーザーはアプリケーションを開始できません。この修正を適用するには、SSL SDKの新しいバージョンをインストールしてください。

[#LC2729]

- セッションの起動に失敗し、次のエラーメッセージが表示されることがありました：

「システムに接続されたデバイスが機能していません。」

これは、ユーザーとサーバーが別々のドメインの場合に、双方向に信頼関係があるドメインのKerberos保護環境で発生します。

[#LC2762]

- Windows Phone 8.1でInternet Explorerの公開されたバージョンを開くと、デバイスが横向きの場合、画面の半分が空白のウィンドウとして表示されるという問題がありました。Internet Explorerウィンドウを最大化することもできません。デバイスを縦向きにすると、画面は正常に表示されます。

[#LC2786]

- セッション共有および残留セッションが有効になっていると、切断されたセッションが同じVDAに再接続できず、異なるVDAに再接続することがありました。この問題が発生した場合、イベントID 1048、イベントID 1050、およびイベントID 7がVDAに記録されます。

[#LC2860]

- 公開アプリケーションが動作を停止すると、Desktop Window Managerによりユーザーがアプリケーションを閉じることができなくなることがありました。

[#LC2922]

- ユーザーがサーバー上のセッションに接続すると、ようこそ画面以降にセッションが進まないことがありました。サーバーが再起動されるまで、この問題はサーバー上のすべてのセッションで発生します。

[#LC2936]

- VDAのHKEY\_CURRENT\_USER\Control Panel\Desktop\UserPreferencesMaskにあるレジストリキーと値が、VDAにログオンするたびにwfshell.exeプロセスにより上書きされるという問題がありました。この問題を避けるには、VDA側に以下のレジストリキーを設定し、値を1にします。

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix

DWORD値を設定します：

EnableVisualEffect

[#LC2965]



- デスクトップおよびダウンロードのショートカットが削除された後、ユーザーが再度VDAにログインすると、%userprofile%\Linksの下のWindows Exploreのお気に入りメニューにそのショートカットが再作成されました。

[#LC2998]

- ユーザーがログオンとログオフを繰り返すと、Citrix WinFrame Shell (Wfshell.exe) が応答を停止します。

[#LC3003]

- ユーザーがReceiver 4.2でログオンすると、セッションで表示される時間がユーザーデバイスの時間とは一致しません。

[#LC3067]

- 管理者がSystem32フォルダーから公開されているアプリケーションを起動しようとするとう失敗します。

[#LC3148]

- ラップトップで外部モニターを使用している場合、外部モニター画面の隅に白い矩形が表示されることがあります。この問題は、とりわけ外部モニターの画面解像度がプライマリモニターの解像度と違う場合に起こります。矩形が表示されるのはプライマリモニターの左側と、プライマリモニターの上端です。

[#LC3186]

- 負荷評価基準を使用して負荷限界状態のサーバーをオフラインにした後、切断セッションに接続できなくなることがありました。

[#LC3226]

- Receiver 4.2から公開デスクトップに接続すると、Citrix WinFrame Shell (wfshell.exe) が失敗します。

[#LC3227]

- XenAppサーバー上の公開アプリケーションまたはデスクトップを起動すると、シングルサインオンが機能しなくなることがありました。

[#LC3301]

- USBのリダイレクトを無効にして、公開アプリケーションから切断すると次のエラーが発生し、イベントビューアーに偽のイベント261が表示されます。

「Citrix Device Redirector ServiceはRedirector BusでIOオペレーションを完了できませんでした。」

[#LC3439]

- 64ビットのThinAPPパッケージかアプリケーション (BMC Control-M) を終了すると、アプリケーションがsfhook64.dllで予期せず例外となることがあります。

[#LC3484]

- まれに、切断セッションに再接続した後クリップボードのリダイレクトに失敗することがあります。再接続時にクリップボード仮想チャネルが予期せず閉じると、この問題が発生します。

[#LC3511]

- VDAでSSL接続 (HTTPS) が正しく動作しないことがあります。

[#LC3535]

## サイト/ファームの管理

- レジストリ HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\Logon\DisableStatus の値を 1 (有効) に設定すると、End User Experience Monitoring (EUEM) メトリクス [ログオンスクリプト実行サーバーの持続時間] と [プロファイルのロード時のサーバーの持続時間] が表示されなくなることがありました。

[#LA5764]

- 負荷評価基準設定のチェックに失敗し、以下のエラーメッセージが表示されることがありました。

「System.Runtime.InteropServices.COMException (0x80004005) : エラー HRESULT E\_FAIL が COM コンポーネントの呼び出しから返されました。」

[#LC0359]

- ポリシーの [再起動スケジュールのランダム化間隔] の設定が無視され、複数の XenApp サーバーが同時に再起動されることがありました。

[#LC1402]

- リモートデスクトップゲートウェイが指定されると、非管理者ユーザーは XenApp 7.6 サーバーに対する RDP セッションを確立できません。

[#LC2371]

- Microsoft DsBind() API 関数に失敗した後、サーバーのローカルホストキャッシュに対して偽の更新が実行されます。この問題の現象として、Controller 上でのローカルホストサイズの急激な増加および Active Directory OU をポイントしているワーカーグループの内容の変更があります。

[#LC2631]

- 'Farm Servers' オブジェクトを含まないワーカーグループを検証すると、"dscheck /full workergroup" コマンドが間違っエラーを報告することがあり、次のエラーが表示されます：

「エラー：WorkGroupUID のサーバー UID を取得できませんでした」

この修正により、この環境で表示されるメッセージが次のよう変更されます。

「このワークグループは Farm Server 用に構成され、サーバー UID がなく、検証に問題はありません。」

[#LC3383]

- Provisioning Services でプロビジョニングされたサーバーで [再起動スケジュールのランダム化間隔] 設定が正しく適用されないという問題がありました。

[#LC3430]

- Worker Group にアプリケーションを公開する場合、コマンド「dscheck /full apps /servercheck」を実行すると不明のサーバーとして Worker Group オブジェクトが間違っ返され、削除するオプションが提供されます。

[#LC3556]

## スマートカード

- Citrix Smart Card ServiceによるCPU使用率が異常に高くなることがありました。この問題を解決するには、Citrix Smart Card Serviceを再起動する必要がありました。

[#LA5919]

- VDAのアップグレード後、スマートカードを使用してMicrosoft Outlookにログオンしようとする時、暗証番号 (PIN) の入力が求められません。この場合、次のようなメッセージが表示されます。

「スマートカードを挿入してください。」

[#LC0556]

- スマートカードを使用してサーバーにログオンすると、「ようこそ」の画面でサーバーが応答不能になり、新しいセッションを開始できなくなることがありました。この場合、Smart Card Serviceを再起動するとセッションを開始できるようになります。

参照番号#LA0983がインストール済みで、以前のバージョンのHotfixインストール時にレジストリを設定した場合、次のレジストリキーの設定は必要ない場合があります。

HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\SmartCard

値の名前: TransactionTimeoutEnable

種類: REG\_DWORD

値の名前 - Enable

HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\SmartCard

値の名前: TransactionTimeoutValue

種類: REG\_DWORD

それ未満の値はすべてデフォルトの5秒となります。

HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\SmartCard

値の名前: SendRecvTimeout

種類: REG\_DWORD

HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\SmartCard  
名前: SendRecvTimeout  
種類: REG\_DWORD  
値: 30秒以上の最小タイムアウト値 (秒)。それ未満の値はすべてデフォルトの30秒となります。10秒以上でTransactionTimeoutValueよりも大きい値を指定してください。

[#LC0910]

- スマートカードリーダーにカードを挿入した状態でCitrix Receiver for Linuxのエンドポイントからログオンすると、セッションが応答不能になることがありました。

[#LC0982]

- scardhook64.dllで発生する例外により、ActiVIdentityプロセス (accompkcs.exe) が応答不能になることがありました。この問題が発生すると、スマートカードで認証するWebサイトを使用できなくなり、VDAの再起動が必要になります。

[#LC1056]

- Citrix Smart Card Serviceが応答不能になり、サービスが再起動されるまでユーザーがサーバーにログオンできなくなることがありました。

[#LC1276]

- セッション中にリーダーからスマートカードを取り出すと、サーバー上でセッションを適切に終了できません。その結果、

これ以降いずれのセッションに対してもスマートカードログオンを実行できない状態であることがサーバーに示されます。

[#LC3020]

- スマートカードを使って公開デスクトップに接続すると、Citrix Smart Card Serviceが応答しなくなることがありました。

[#LC3421]

#### システムの例外

- サーバーのpicaser.sysで、バグチェックコード0x0000000Aによるブルースクリーンエラーが発生することがありました。

[#LC0222]

- サーバーのpicadm.sysで、バグチェックコード0x22によるブルースクリーンエラーが発生することがありました。

[#LC0227]

- XML仲介サーバーとして動作するXenApp 6.5サーバーにHotfix Rollup Pack 3または4をインストールすると、Remote Desktop Servicesが異常停止することがありました。

[#LC0440]

- サーバーのvdtw30.dllでブルースクリーンエラーが発生することがありました。

[#LC0471]

- サーバーのpicadm.sysで、バグチェックコード0x00000022によるブルースクリーンエラーが発生することがありました。

[#LC0483]

- 参照番号#LA5282に対する修正 (XA650W2K8R2X64R04 - XenApp 6.5 Hotfix Rollup Pack 4などに含まれています) が適用されたシステム上のセッションで、公開アプリケーション上のURLをクリックしたときにwinlogon.exeプロセスが異常停止することがありました。この結果、その公開アプリケーションが切断されます。

[#LC0523]

- Citrix XML Serviceが予期せず失敗することがありました。

[#LC0548]

- パススルーセッションのMedia Playerでビデオを再生すると、セッションが予期せずに終了する場合があります。

[#LC0553]

- サーバーのvdtw30.dllで、バグチェックコード0x0000007Eによる重大な例外が発生し、ブルースクリーンに表示されることがありました。

[#LC0599]

- オペレーティングシステムでpicadm.sysエラーが発生し、ブルースクリーンが表示されます。

[#LC0623]

- サーバーのpicadm.sysで、バグチェックコード0x0000000Aによるブルースクリーンエラーが発生することがありました。

[#LC0649]

- リモートデスクトッププロトコル (RDP) 接続からのログオフ時に、winlogon.exeプロセスがtwi3.dllで異常終了することがありました。  
[#LC0793]
- VDAのpicadd.sysでブルースクリーンエラーが発生することがありました。  
[#LC0798、#LC1096]
- VDAのctxdvc.sysでブルースクリーンエラー (バグチェックコードE3) が発生することがありました。  
[#LC0857]
- オペレーティングシステムのpicadmsysで、停止コード0x0000003Bによるブルースクリーンエラー発生することがありました。  
[#LC0896]
- 以下のいずれかの場合、Windowsエクスプローラーは予期せずに閉じてしまうことがあります。
  - 名前に260文字を越える文字が含まれているファイルを大量に選択して、[送る]の[Fax受信者]オプションを選択した場合。
  - サードパーティのアプリケーションを開こうとした場合。
  - Nitro PDFを使用して、ファイルを結合しようとした場合。
 [#LC0938]
- ICA WSK Transport Driver (tdwsk.sys) のページフォールトにより、VDAでブルースクリーンエラーが発生します。  
[#LC1032]
- 公開されたアプリケーションやデスクトップの起動時にicaendpoint.dllでWindowsログオンユーザーインターフェイス (LogonUI.exe) またはexplorer.exeのプロセスが応答不能になることがありました。この問題は、Citrix Audio Redirection Serviceが起動に失敗すると発生します。  
[#LC1167]
- オペレーティングシステムでpicadmsysエラーが発生し、ブルースクリーンが表示されます。  
[#LC1239]
- サーバーのvdtw30.dllが予期せず終了することがありました。  
[#LC1328]
- サーバーのvdtw30.dllでブルースクリーンエラーが発生することがありました。  
[#LC1414]
- twi3.dllのエラーによりサーバーが応答を停止することがあります。  
[#LC1417]
- ターミナルサービスのsvchost.exeプロセスが予期せず異常終了してアプリケーションイベントID 1000が発生し、サーバーへのReceiver接続およびリモートデスクトップ接続の確立に失敗することがありました。  
[#LC1419]

- オペレーティングシステムでpicadm.sysエラーが発生し、ブルースクリーンが表示されます。この問題は、TrucationResourceがnullになる場合に発生します。

[#LC1430]

- vdtw30.dllで、バグチェックエラー0x0000003B, {c0000005, fffff96000863133, fffff880099aaf70, 0}によるブルースクリーンエラーが引き起こされることがあります。

[#LC1567]

- サーバーのpicadm.sysでデッドロックが発生することがあり、それによりユーザーがサーバーに接続できなくなることがありました。

[#LC1685]

- 無効なセッションキーがあるとCitrix Stack Controlサービスが予期せず終了します。

[#LC1717]

- ICAセッションでSSLが有効になっている場合、Receiver for HTML5からオーディオファイルを再生するとVDAがブルースクリーンのエラーで失敗することがありました。

[#LC1752]

- ICA Audio Endpoint dll (icaendpoint.dll) が原因で、svchost.exeプロセスが予期せず停止することがありました。

[#LC1775]

- まれに、VDAが予期せずシャットダウンして、再起動することがあります。

{#LC1876}

- ターミナルサービスがRPM.dllにより異常終了することがありました。

[#LC2031]

- サーバーのvdtw30.dllで、バグチェックコード0x0000003Bによる重大な例外が発生し、ブルースクリーンに表示されることがありました。

[#LC2037]

- オペレーティングシステムのctxdvc.sysで、バグチェックコード3Bによるブルースクリーンエラーが発生することがあります。

[#LC2119]

- オペレーティングシステムのtdica.sysで、バグチェックコード44[MULTIPLE\_IRP\_COMPLETE\_REQUESTS]によるブルースクリーンエラーが発生することがあります。

[#LC2171]

- Windows Server 2008 R2およびWindows Server 2012 R2では、Receiver for Windowsで接続すると重大なエラーが発生することがありました。

[#LC2179]

- picadms.sysで、バグチェックコード0x00000022によるブルースクリーンエラーが発生することがありました。  
[#LC2236]
- twi3.dllで、winlogon.exeプロセスが予期せず閉じることがありました。  
[#LC2244]
- オペレーティングシステムのpicadms.sysで、バグチェックコード0x00000022によるブルースクリーンエラーが発生することがありました。  
[#LC2265]
- VDAサーバーのicardd.dllでブルースクリーンエラーが発生することがありました。  
[#LC2277]
- XML仲介サーバーとして動作するXenApp 6.5サーバーにHotfix Rollup Pack 5をインストールすると、Remote Desktop Servicesが予期せず停止することがありました。  
[#LC2376]
- サーバーのctxdvc.sysで、バグチェックコード0x0000003Bによる重大な例外が発生し、ブルースクリーンに表示されることがありました。  
[#LC2444]
- 公開デスクトップセッションの実行中にWindows Explorerの処理が予期せず中断し、「Windows Explorerが動作を停止しました」という意味のメッセージが表示されることがあります。  
[#LC2733]
- COMポートを経由してPOSデバイスが共有デスクトップにリダイレクトされると、共有デバイス上で実行中のサードパーティソフトウェアからデバイスが認識されないことがありました。  
[#LC2775]
- サーバーのicausb.sysでエラーが発生し、ブルースクリーンにメッセージPNP\_DETECTED\_FATAL\_ERRORが表示されることがありました。  
[#LC2811]
- wfshell.exeプロセスが異常終了し、新規の接続ができなくなることがありました。  
[#LC3024]
- サーバーでブルースクリーンにエラーが表示され、vdtw30.dllおよびwin32k.sysで重大な例外が発生することがあります。  
[#LC3103]
- 「client clipboard redirection」を有効にすると、wfshell.exeプロセスが予期せずに終了していました。  
[#LC3249]
- VDAのvdtw30.dllでブルースクリーンエラーが発生することがありました。

[#LC3354]

- csrss.exeプロセスで、停止コード0x7eによりvdtw30.dllで重大な例外が発生することがありました。

[#LC3482]

#### ユーザーエクスペリエンス

- LAN接続でReceiver for Windows Version 4.200よりも前のバージョンを使用すると、デスクトップコンポジションリダイレクト機能で操作上の問題が生じることがありました。

[#LC1507]

- このリリースには、Receiver for HTML5またはReceiver for Chromeでログオンするユーザーが、リモートセッションおよびローカルデバイスからファイルを転送できるようになる拡張が含まれます。

[#LC3014]

#### ユーザーインターフェイス

- この修正により、Receiver for Windowsのタスクバーでのアプリケーションの表示に関する以下の問題が解決されます。

- アプリケーション固有のアイコンではなくReceiverのデフォルトアイコンが表示される。
- 同一セッション内で複数のアプリケーションを実行する場合にReceiverの単一のデフォルトアイコンにグループ化（結合）される。

[#LC0094、#LC1181]

- 「コンボボックスをデバイス側で表示する」ポリシーが有効な場合、ユーザーがiPadを使ってReceiver for iOSでInternet Explorerを開くと、リモートコンボボックスのバルーンが正しく表示されませんでした。たとえば、コンボボックスBをクリックすると、バルーンがコンボボックスAに表示されます。

[#LC1800]

- シームレスアプリケーションウィンドウの特定の境界線が表示されないことがありました。この問題は、「GetWindowRect」関数が正しいウィンドウサイズを取得できない場合に発生し、その結果ウィンドウ座標が境界線なしで表示されます。

[#LC1943]

- Receiver for WindowsまたはモバイルReceiverから公開アプリケーションを開くと「キーボードの自動表示」が表示されないことがありました。

[#LC2747]

#### その他

- この修正により、Microsoft社の更新プログラムとの互換性が向上します。

[#LA5638]

- この修正は、基本コンポーネント内のメモリ問題に対応しています。

[#LA5765]

- ライセンスサーバーアドレスを取得しようとする、ライセンスサーバー名が正しい文字で表示されないことがありまし



た。

[#LC0363]

- WMIプロセスでハンドルリークが発生することがありました。

[#LC0662]

- ユーザーフォルダーのリダイレクト機能で、205文字を超える名前のファイルの処理に問題が発生することがありました。

[#LC0719]

- この修正は、基本コンポーネント内のメモリ問題に対応しています。

[#LC0732、#LC0800、#LC0953、#LC2764、#LC2888]

- 公開アプリケーションのExcelを実行してから、[電子メールで送信]を使ってファイルを送信する場合、ユーザーがOutlookで文字列を入力してBackSpaceキーを数回押すと、バックスペースの代わりに漢字が挿入されました。

[#LC1155]

- ユーザーがReceiver for Macでログオンすると、USBアダプタ付COMポート署名パッドは正常に機能しません。ユーザーがReceiver for Windowsでログオンすると、署名パッドは正常に機能します。

[#LC1396]

- CTXCDFのハンドルリークによってWindows Management Instrumentationプロバイダサービス (WMIPRVSE.exe) が停止し、イベントログにイベントID 5612が記録されることがありました。

[#LC1764]

- この修正プログラムは、Citrixオーディオリダイレクトサービス (CtxAudioSvc) でのメモリリークを解決します。

[#LC2054]

- Powershellコマンドの"Get-BrokerSession | Format-List -Property SessionState"が正しくない折衝状態を示します。たとえば、切断された接続がアクティブなセッションとして表示されます。

[#LC2080]

- このリリースには、新しいICA仮想チャネルであるFramehawk仮想チャネルのサポートが含まれます。Framehawk仮想チャネルはCitrix HDXテクノロジーを拡張し、パケット損失と遅延の問題がよく発生するブロードバンドワイヤレス接続でユーザーエクスペリエンスを向上させます。詳しくは、「[新機能](#)」を参照してください。

[#LC2782]

- この拡張により、イベントID 1480のメッセージが向上し、またこれには次の情報が含まれています。

- 公開アプリケーション名
- コマンドライン
- User ID
- ユーザードメイン
- クライアント名

- セッションID

[#LC2830]

- VDAのC:ドライブのルート上でのsessionchange.logファイルの作成は無効にできず、別の場所に移動できません。

[#LC3621]

# 既知の問題

Jan 04, 2017

## Feature Pack 3

- 最近のバージョンのFirefoxでは、URLリダイレクト用のブラウザアドオンはデフォルトで無効になっているため、これを有効にする必要があります。これはサードパーティ製の問題です。[#283844]
- ユーザーがWindows 10 VDA内でファイルをEdgeブラウザーでダウンロードした場合、DownloadsおよびDesktopなどのリダイレクトされたフォルダーから一部のファイルを開くことができません。この問題を解決するには、Internet Explorerを使ってファイルをダウンロードします。[#591635]
- ブローカーレスVDAをFP3にアップグレードした後は、次のレジストリキーが存在せず、手動で追加する必要があります[#591030]:
  - hklm\software\citrix\VirtualDesktopAgent\HaModeComputerName
  - hklm\software\citrix\VirtualDesktopAgent\HaModeTimeEnd
- Windows 10 VDA上でWacomペンデバイスとReceiver for Windowsを使用する場合、ネットワーク接続がなくなると、(クライアント自動再接続を使って)セッションに再接続してもカーソルが表示されないことがあります。この問題を解決するには、そのセッションを切断して再接続してください。[#593516]
- Citrix Telemetry Serviceを実行している場合、永続的CDFトレースは使用できません。この問題を回避するには、Citrix Telemetry Serviceを一時的に無効にします。[#593347]
- XenServer上で既存のWindows 10マスターイメージを更新して、XenServer HotfixXS65ESP1010とXenToolsを使用すると、XenAppおよびXenDesktop 7.6 FP3でエラーが発生します。[#593985]
- Windows 10 VDAから公開されるアプリは、最大化した時に完全な境界線を表示しません[#593596]。この問題を解決するには、各アプリケーションクラスにレジストリキーを設定します。たとえば、ペイントの動作について問題を解決するには、次のキーを作成します:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshell\TWI\MSPaintApp]
```

```
"ClassName"="MSPaintApp"
```

```
"Type"=dword:00020000
```

アプリケーションにはアプリケーション名ではなく、クラス名を追加する必要があります。たとえば、ペイント (Paint) に **MSPaintApp** を追加します。クラス名は、Spy++などのツールを使って取得できます。シームレス構成設定について詳しくは、「[CTX101644](#)」を参照してください。

- 「詳細情報を表示する」リンクが現在の製品ドキュメントサイト (docs.citrix.com) に正しくリダイレクトされないことがあります。この問題を解決するには、製品ドキュメントサイトに直接アクセスしてください。7.6 FP3およびそれより前の7.6 Feature Packsのドキュメントはすべて、新しい製品ドキュメントサイトの「[XenAppおよびXenDesktop 7.6](#)」にあります。
- インストールが完了した後、スタンドアロンVDAインストールパッケージの一時ファイルはVDAオペレーティングシステムの一時フォルダーに残ります。この問題を解決するには、一時ファイルを手作業で削除してください。[#594829]

## Feature Pack 2

セッション内でテキストをコピーしてセッション外のアプリケーションに貼り付ける場合、およびセッション外からコピーしてセッション内に貼り付ける場合に、リッチテキストの書式設定が失われます。[#0538497]

ホスト名 (完全修飾ドメイン名 (FQDN) のホストの部分) の15文字を超える部分が切り捨てられ、VDAの登録に失敗します。[#0544120]

休止状態から再開したときに、Linux VDAの登録状態がエンドポイントで変化しません。電源がオンになるときにエンドポイントでIPアドレスを取得できず、そのために登録に失敗します。[#0569303]

現在、Linux VDAでは領域マッピングのDNS参照はサポートされていません。[#057555]

KDEベースのアプリケーションはVDAセッションで利用不可と表示されます。[#0568006]

Quest Authentication Serviceが有効でSELinuxが強制モードの場合は、Red Hat Enterpriseが動作するLinux VDAで接続に失敗します。[#0551761]

Citrix Receiver for Windows (バージョン4.3) およびCitrix Receiver for iOS Classic (バージョン6.0) は、現在サポートされている唯一のReceiverプラットフォームです。

Session Recording/Smart AuditorはFramehawkでサポートされていません。

このリリースのFramehawkは、ナレッジワーカーおよびオフィスアプリなど、標準のXenAppおよびXenDesktopのワークロード用に設計されています。高度なグラフィック集中環境ではテストされていません。

マルチモニタークライアントは現在サポートされていません。

WANエミュレーターインセッションで帯域幅や待機時間を変更するなどのテスト状況下では、クライアント上のVd3dn.dllがクラッシュすることがあります。WANエミュレーター設定にいずれかの変更を加える前にセッションを切断し、その後で再接続します。Framehawkはセッションハンドシェイク中に再補正します。

Framehawkポリシーおよびレガシポリシーの両方が有効な場合、ワークステーションのオペレーティングシステム (Windows 7など) にはFramehawkポリシーが優先され、RDS環境 (Windows 2012 R2など) の場合は、レガシポリシーが適用されます。

Windows 7環境の場合、Windows 7オペレーティングシステムにタッチサポート (ピンチおよびズームなど) 機能が装備されているにもかかわらず、Framehawkの初期XenDesktop 7.6 FP2リリースではそれを実行できません。また、Windows 2008 R2にはタッチサポートが実装されていません。

StoreFront 3.0の既知の問題および解決された問題については、StoreFront 3.0のドキュメントの [既知の問題](#) および [解決された問題](#) を参照してください。

HDX RealTime Optimization Pack 1.8の既知の問題および解決された問題については、Optimization Pack 1.8のドキュメントの [既知の問題および解決された問題](#) を参照してください。

# Feature Pack 1

- Machine Creation Services (MCS) または Provisioning Services で、構成済みのマスターイメージとインストール済みの Microsoft Message Queuing (MSMQ) を使用して VDA を作成すると、その VDA の QMid は MSMQ と同じになります。これは、次のようなさまざまな問題が発生する原因となる場合があります。
    - 録画契約が承認されていても、セッションが録画されない場合があります。
    - セッションのログオフ信号が Session Recording Server によって受信されず、セッションが常にライブステータスになってしまう可能性があります。[#528678]
- VDA ごとに固有の永続的な QMid を作成するための解決策は、スクリプトを使用することです。スクリプトを使用するには、以下の手順に従ってください。
- PowerShell の実行ポリシーが RemoteSigned か Unrestricted に設定されていることを確認します。  
**Set-ExecutionPolicy RemoteSigned**
  - スケジュールされたタスクを作成し、トリガーを At system startup に設定して、Provisioning Services または MCS マスターイメージマシンで SYSTEM アカウントを使って実行します。
  - スタートアップタスクとしてコマンドを追加します。  
**powershell.exe -file C:\GenQMID.ps1**

## 警告

レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windows の再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrix では一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

スクリプトの概要：

- マシンの FQDN のハッシュ値に基づいて QMid を生成します。
- CitrixSmAudAgent や MSMQ などの関連サービスを停止します。
- レジストリで QMid を設定します。
- 以前に停止したサービスを開始して、QMid の変更を適用します。

```
function ConvertHexStringToByte($theString)
{
    $bytes = New-Object Byte[] ($theString.Length / 2)
    for ($i = 0; $i -lt $theString.Length; $i += 2) {
        $bytes[$i / 2] = [System.Convert]::ToByte($theString.Substring($i, 2), 16)
    }
    return $bytes
}

Try {
    # Get UUID of machine
    $strUUID = (Get-WmiObject -Class Win32_ComputerSystemProduct | Select-Object -Property UUID).UUID

    # Remove "-"
    $strUUID = $strUUID.ToString().Replace("-", "")

    # Convert string to bytes
    $UUID = ConvertHexStringToByte($strUUID)

    # Set UUID as QMid
    $new_QMID = $UUID
} Catch {
    # IF exception occurred, just use MD5 digest of FQDN as QMID
```

```

# Get FQDN

$fqdn = [System.Net.Dns]::GetHostByName(($env:computerName)).HostName

# Calculate MD5 hash of FQDN

$md5 = new-object -TypeName System.Security.Cryptography.MD5CryptoServiceProvider

# Set md5 digest as QMID

$utf8 = new-object -TypeName System.Text.UTF8Encoding

$new_QMID = $md5.ComputeHash($utf8.GetBytes($fqdn))
}

# Write new_QMID into registry

Set-ItemProperty -Path HKLM:\Software\Microsoft\MSMQ\Parameters\MachineCache -Name "QMID" -Value $new_QMID

# Restart MSMQ to adopt new QMID

# Get dependent services

$depServices = Get-Service -name MSMQ -dependentservices | Select -Property Name

Restart-Service -force MSMQ

# Start dependent services

if ($depServices -ne $null) {

foreach ($depService in $depServices) {

$startMode = Get-WmiObject win32_service -filter "NAME = '$($depService.Name)'" | Select -Property StartMode

if ($startMode.StartMode -eq "Auto") {

Start-Service $depService.Name

}

}

}
}

```

- 4096 x 4096以上の解像度でセッションを録画すると、録画にフラグメントが表示される場合があります。[#524973]
- XenAppまたはXenDesktopのライセンスの種類を変更しても、その変更がSession Recordingにすぐに反映されません。解決策：VDAマシンを再起動します。[#532393]
- 次の2つの場合、「Installation failed」というエラーが表示される場合があります。このメッセージは無視できますが、表示されないようにするには、Session Recordingコンポーネントを再インストールする前にマシンを再起動します。[#544579]
  - Session Recordingコンポーネントをアンインストールしてから、マシンを再起動せずに再インストールしました。
  - インストールが失敗してロールバックが発生したため、マシンを再起動せずSession Recordingコンポーネントを再インストールしようとしました。
- Session Recordingの事前起動アプリケーションセッションのサポートには制限があります。[BUG0561109]
  - 問題：
    - アクティブなポリシーによりアプリケーション名のマッチングが試行されるとき、事前起動セッションで起動されたアプリケーションはマッチングされません。そのため、セッションは録画されません。
    - アクティブなポリシーですべてのアプリケーションを録画するように指定している場合、ユーザーがWindows Receiverに（事前起動セッションが確立されると同時に）ログオンすると、録画の通知が表示され、空のセッションおよびこのセッションで後から起動されるアプリケーションが録画されます。
  - 回避方法：
    - アプリケーションを録画ポリシーに従って別々のデリバリーグループに公開します。録画条件としてアプリケーション名を使用することを避けます。これにより事前起動セッションが確実に録画されます。ただし、それでも通知は表示されます。
- 4096 x 4096以上の解像度でセッションを録画すると、録画にフラグメントが表示される場合があります。[#524973]
- XenAppまたはXenDesktopのライセンスの種類を変更しても、その変更がSession Recordingにすぐに反映されません。解決策：VDAマシンを再起動します。[#532393]
- 次の2つの場合、「Installation failed」というエラーが表示される場合があります。このメッセージは無視できますが、表示されないようにするには、Session Recordingコンポーネントを再イン：

- Session Recordingコンポーネントをアンインストールしてから、マシンを再起動せずに再インストールしました。
  - インストールが失敗してロールバックが発生したため、マシンを再起動せずSession Recordingコンポーネントを再インストールしようとしました。
- 4096 x 4096以上の解像度でセッションを録画すると、録画にフラグメントが表示される場合があります。[#524973]
  - XenAppまたはXenDesktopのライセンスの種類を変更しても、その変更がSession Recordingにすぐに反映されません。解決策：VDAマシンを再起動します。[#532393]
  - 次の2つの場合、「Installation failed」というエラーが表示される場合があります。このメッセージは無視できますが、表示されないようにするには、Session Recordingコンポーネントを再インストールしてください。
    - Session Recordingコンポーネントをアンインストールしてから、マシンを再起動せずに再インストールしました。
    - インストールが失敗してロールバックが発生したため、マシンを再起動せずSession Recordingコンポーネントを再インストールしようとしました。
  - 4096 x 4096以上の解像度でセッションを録画すると、録画にフラグメントが表示される場合があります。[#524973]
  - XenAppまたはXenDesktopのライセンスの種類を変更しても、その変更がSession Recordingにすぐに反映されません。解決策：VDAマシンを再起動します。[#532393]
  - 次の2つの場合、「Installation failed」というエラーが表示される場合があります。このメッセージは無視できますが、表示されないようにするには、Session Recordingコンポーネントを再インストールしてください。
    - Session Recordingコンポーネントをアンインストールしてから、マシンを再起動せずに再インストールしました。
    - インストールが失敗してロールバックが発生したため、マシンを再起動せずSession Recordingコンポーネントを再インストールしようとしました。

Citrixライセンスサーバー11.12.1の既知の問題および解決された問題については、Citrixライセンスサーバー11.12のドキュメントの [既知の問題および解決された問題](#) を参照してください。

HDX RealTime Optimization Pack 1.7の既知の問題および解決された問題については、Optimization Pack 1.7のドキュメントの [既知の問題および解決された問題](#) を参照してください。

## XenApp 7.6およびXenDesktop 7.6

レジストリエントリの変更を伴う解決策については、次の点に注意してください。

注意：レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があります。Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

- サーバーOSマシンでは、CreateAnonymousUsersAppツールで一部の匿名ユーザーアカウントがローカルマシンから削除されない場合があります。すべての匿名ユーザーはローカルユーザーであり、Active Directoryユーザーではありません。ツールによりパスワードとプロファイルが削除されるため、アカウント名が削除されていなくてもその匿名ユーザーアカウントは使用できなくなります。削除されずにサーバー上にローカルに残った匿名ユーザーアカウントを削除するには、ユーザーアカウントの管理またはコンピューターマネージャーを使用してください。[#4999679]
- Universal Print ServerのWebサービスに非標準的なHTTP/SOAPポートを構成するには、PowerShellコマンドレットを使ってセッションプリンターポリシーを変更します。グループポリシー設定の構成について詳しくは、「  
—SDKについて」で、「グローバルポリシーSDKの使用法」セクションを参照してください。次のコマンドを実行します。  
Set-ItemProperty LocalGpo:\Computer\Unfiltered\Settings\ICA\Printing\UniversalPrintServer\UpsHttpPort -name Value -Value <portnumber>#268593]

- Hyper-Vホストの一時停止を解除した後、ホストの全体的な状態がMicrosoft System Center Virtual Machine Manager (VMM) に反映されるまでに時間がかかることがあります。これにより、Machine Creation Services (MCS) の使用に問題が生じることがあります。Hyper-Vホストが一時的停止状態であると認識される場合、そのホストは仮想マシンのプロビジョニングに使用されません。また、すべてのホストが一時的停止状態であると認識される場合、カタログの作成に失敗します。この問題を回避するには、親クラスターノードまたはHyper-Vホストノードを手作業で更新してください。また、ホストの環境テストを実行すると、一時停止状態として認識されているホストが別されます。[#285696]

- デスクトップOS上でホストされるアプリケーションを、リモートデスクトッププロトコル (RDP) で仲介することはサポートされていません。[#377108]
- このリリースでは、Windows 8のセッションでクライアント側ドライブマッピングを構成しても、ISOファイルをマウントすることはサポートされません。[#333111]
- ドライバーの問題によりCitrix Universal Print Serverの処理に失敗する場合は、プリンタードライバーの分離機能を使用してください。この機能について詳しくは、MSDN ([http://msdn.microsoft.com/en-us/library/windows/hardware/ff560836\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/hardware/ff560836(v=vs.85).aspx)) を参照してください。[#381460]
- デスクトップエクステンション拡張のポリシー設定が既存のユーザープロファイルまたは管理者プロファイルに適用されないという問題があります。この問題を解決するには、すべての既存のプロファイルを削除してからこのポリシー設定を有効または無効にして、さらにVDAをインストールしたプロファイルを（インストール後に）削除してください。組み込みの管理者アカウントでVDAをインストールした場合は、プロファイルを削除できません。この場合、ユーザーはログイン時にCitrix Enhanced Desktop Themeを選択できます。[#363736]
- VDAデスクトップセッションの解像度がクライアント側モニターの解像度よりも高いために正しく表示されない場合は、そのモニターでサポートされる最高の解像度をVDAデスクトップに設定してください。または、VDA側のグラフィックカードのモニターケーブルを抜いてください。[#365877]
- スクリーンセーバーおよび節電オプションは、セッションでは無効になります。これを有効にするには、以下のレジストリ値 (DWORD) を作成してください。：  
HKLM\Software\Citrix\Graphics\SetDisplayRequiredMode = 0  
この変更により、リモートマシンのスクリーンセーバーや節電モードが動作するようになります。節電モードが動作した場合、ユーザーがマウスまたはキーボードを操作するまではリモートセッションが更新されなくなりますが、画面はブランクになりません。[#380550]
- ホスト管理アドレスを設定する前にリモートマシンからPowerShell SDKのNew-ProvSchemeコマンドまたは任意のMCSコマンドを使用しようとすると、エラーが生じることがあります。この問題を回避するには、Set-HypAdminConnectionコマンドで管理アドレスを設定してからNew-ProvSchemeコマンドを実行してください。[#336902]
- Hyper-VホストでMicrosoft RemoteFXのvGPU機能を使用することはサポートされません。RemoteFXのvGPU機能を使用するHyper-Vホストにアクセスするには、RDPを使用してください。[#375577]
- Directorの [ダッシュボード] ページの [失敗したマシン] パネルには、起動に失敗したGPUマシンの数が表示されます。ただし、[フィルター] ページで詳細を表示すると、この情報は表示されません。

この情報は、履歴傾向のグラフで確認することもできます。[#0434722]

- XenServerのリソースプールのサポートされる数よりも多いGPUマシンをプロビジョニングまたは割り当てると、GPUマシンの起動に失敗することがあります。この問題が発生すると、そのマシンに割り当てられたユーザーは、Personal vDiskが使用される場合も含め、StoreFrontから仮想デスクトップにアクセスできません。[#0434505]
- GPUマシンの起動に失敗して、次のエラーメッセージが表示されることがあります。

```
「Exception Failure in PowerOn,PGPU_INSUFFICIENT_CAPACITY_FOR_VGPU」
```

このメッセージは、GPUリソースが不足しているために追加のマシンを起動できない場合に表示されます。[#434509]

- RDPおよびICAプロトコルを使用して複数のユーザーがデスクトップOSマシンにログインし、1人のユーザーが自分のRDPセッションをロックすると、ICAで接続しているユーザーがセッションにログインできなくなります。この問題を回避するには、RDPセッションのユーザーはセッションをロックするのではなく、切断またはログオフする必要があります。[#392311]

- Windows 8 VDAホストでアプリケーションを公開し、そのアプリケーションが通知領域に非ASCII文字を表示する場合、再接続後にほかの無関係な文字も通知領域に表示されることがあります。の問題を解決するには、いったんセッションからログオフしてから、再度アプリケーションを起動してください。[#387963]
  - このリリースでは、セッション共有をレジストリで無効にすることはサポートされていません。このため、セッション共有は常時有効になります。レジストリキーでセッション共有を無効にすると、最初のアプリケーションを起動した後はほかのアプリケーションを起動できなくなります。現在、この問題を回避する方法はありません。[#383718]
  - Windows Server 2012上のセッションにユーザーが再接続するときに、デスクトップウィンドウマネージャーでメモリーリークが発生することが確認されています。社員が会社と自宅から同じセッションに接続する場合など、長期間アクティブなままのセッションに対して接続および切断を頻繁に行くと、サーバー上のメモリーリソースが減少し、応答が遅くなったりサーバー障害が発生したりすることがあります。これはサードパーティ (Microsoft社) の問題です。詳しくは、<http://support.microsoft.com/kb/2855336> (2013年7月公開) を参照してください。[#374261]
  - Windows PowerShell 3.0を使用する場合、バージョン情報に関するトピックが開きません。これはサードパーティ (Microsoft社) の問題です。ほかのヘルプトピックは、正しく開きます。[#408866]
  - XenDesktop 5からこのリリース (またはXenDesktop 7.0や7.1) にアップグレードするときに、共有およびプライベートのデリバリーグループの両方に割り当てられたアプリケーションでデリバリーグループ非互換エラーが発生します。この問題を回避するには、ホストされるアプリケーションを共有およびプライベートのデリバリーグループの両方に割り当てないでください。[#41942]
  - Profile management機能が有効な場合、Windows Server 2012 R2またはWindows 8.1のセッションでのログオンスクリプトの実行は、デフォルトで5分遅れます。この場合、セッションの起動が完了したときに、Directorにログオン期間の情報が正しく表示されません。ログオンスクリプトの遅延は、[Configure Logon Script Delay] ポリシー (有効: 0) により制御されます。[#407978]
  - Microsoftのソフトウェアの更新管理がインストールされたWindows 8.1が動作するコンピューター上で、デスクトップを起動できないことがあります。この問題を回避するには、Windows 8.1の [タスクバーとナビゲーションのプロパティ] ダイアログボックスの [ナビゲーション] タブで [スタート画面] にあるすべてのチェックボックスをオンにしてください。[#408439]
  - 仲介されるセッションへのRDP接続で、「使用できるライセンスがないため、このセッションにアクセスできません」というエラーメッセージが表示されることがあります。この問題を回避するには、C:\inetpub\wwwroot\Citrix\Store\App\_Data\default.icaで以下の設定を無効にしてください。[#422212, #403855]
    - RDPConnection=false
    - RDP-RedirectDrives=false
    - RDP-RedirectDynamicDrives=false
  - Studioの [ポリシー] ノードで、[固定キャッシュしきい値] 設定の単位が「Kbps」として表示されるという問題があります。正しくは、ビット/秒 (bps) です。[#429478]
  - 構成ログの日付の範囲として [カスタム] を選択する場合、カレンダーが正しく表示されないことがあります。[#452399]
  - Amazon Web ServicesまたはCloudPlatform上でプロビジョニングされたマシンが正しく一時停止状態にならないことがあります。この問題は、そのデリバリーグループの電源管理設定で、切断に一時停止するように構成しても発生します。[#453780]
  - Windows Server OS VDAがインストールされたWindows Server 2012マシンで接続タイマー (最长接続タイマー、接続タイマー、および切断タイマー) で問題が発生し、セッションで予期されないタイムアウトが発生することがあります。[#471698]
  - ポリシーの [汎用ユニバーサルプリンターを自動作成する] が有効な場合に、Citrix Print Manager Service (CpSvc.exe) プロセスで汎用ユニバーサルプリンターが作成されないという問題がありました。この問題を解決するためのHotfixがCitrix Knowledge CenterのCTX141565およびCTX141566で提供されています。
  - Citrixポリシーの [許可される最大表示色数] が有効な場合、Windows Server 2012 R2を実行するVDAへのユーザー接続が失敗する可能性があります。Windows Server 2012 R2を実行するVDAのような、プライマリディスプレイドライバーとしてWindows Display Driver Model (WDDM) ドライバーを使用するVDAには、[許可される最大表示色数] が適用されません。
- 
- Citrixメタインストーラーを使用してVirtual Delivery Agent (VDA) をインストールまたはアップグレードすると、「Error 1904. Module C:\Program Files (x86)\Citrix\System32\rpm.dll failed to register. HRESULT -2147010895. Contact your support personnel (エラー1904. モジュールC:\Program Files (x86)\Citrix\System32\rpm.dllで登録に失敗しました。HRESULT -2147010895。サポート担当者にお問い合わせください)」が発生して、サードパーティ製ツールMicrosoft Visual C++ 2005 Runtimeコンポーネントのインストールが失敗し、インストールがロールバックされるにもかかわらずインストールが正しく完了したと認識されることがあります。この問題が発生すると、残りのVDAコンポーネントのインストールが続行され、インストール後のコンポーネントの初期化処理まで到達してしまいます。この段階で、Microsoft Visual C++ 2005 Runtimeコンポーネントがインストールされていないため、続行できなくなります。この問題を解決するには、メタインストーラーによるVDAのインストールを再実行してください。これにより、Microsoft Visual C++ 2005 Runtimeコンポーネントが正しくインストールされます。または、Microsoft社の<http://support.microsoft.com/?kbid=947821>の手順に従ってください。[#489633]
  - サーバーオペレーティングシステムへのVirtual Delivery Agentのインストールに失敗して、エラー1935が発生することがあります。この問題は、Microsoft Visual C++ 2005再頒布可能パッケージの後方互換性エラーにより発生します。この問題を解決するには、Microsoft社のWebサイトを参照するか、Windows Updateを実行して修正プログラムを入手してください。[#354833]
  - VDA for Windows Server OSのインストールに成功した後、マシンを再起動する前に、Windowsイベントログに複数のエラーメッセージが記録されることがあります (ターミナルサーバーリスナースタックがダウンしているか、セッションの作成に失敗したことを示すTemService 1035または1036など)。ほかにインストールエラーを示すものがない場合は、これらのイベントログメッセージを無視することができます。[#374134]
  - XenDesktop 5.xからアップグレードする場合は、XenDesktop 5.x Desktop Studioを閉じておく必要があります。Desktop Studioを実行しておくと、アップグレード中にDesktop Studioが異常終了することがあります。[#389374]
  - XenDesktop 5.6からのアップグレードで、無効に設定されていた管理者が役割やスコープなしのままで移行されることがあります。アップグレード後にStudioで管理者の表示を確認して、必要な場合は管理者を編集してください。[#394765]
  - Profile management機能が有効な場合、初回ログオン時に実行されるデフォルトのWindows 8アプリケーション (天気、ニュース、Bingなど) が、以降のログオン時に実行されなくなることがあります。この問題は、ユーザープロファイルが存在せず、AppData\Local\フォルダーが除外されていない (デフォルト) 環境で発生します。この問題を解決するには、Profile managementを再構成して、AppData\Local\PackagesおよびAppData\Local\Microsoft\Application Shortcutsフォルダーを除外してください。[#394802]
  - Windows 7上でVDAのインストールまたはアップグレード中に、コンピューターの再起動を求めるWindowsのダイアログボックスが表示されることがあります。この場合は [後で再起動する] をクリックしてアップグレードを続行してください。 [今すぐ再起動する] を選択しないでください。[#396553]
  - Active Directoryグループポリシーで個々のMSIファイルを使ってVDAをインストールすることは推奨されず、失敗する可能性があります。Active Directoryのスクリプトを使ったVirtual Delivery Agentのインストールまたは削除の説明に従って、インストールメディアに収録されているスタートアップスクリプトを使うことをお勧めします。[#383432, #372136]
  - VDAのインストールで、最適化に時間がかかることがあります。一部のテストでは30分かかり、それ以上かかる場合もあります。この問題は、Windowsを実行するイメージ上にVDAをインストールするときに、インストールウィザードの [機能] ページで [パフォーマンス最適化] を選択すると発生します。これにより、Microsoft Native Image Generator (Ngen) が実行されます。この問題が発生した場合は、インストール処理が完了するまでお待ちください。この問題を回避するには、仮想デスクトップをプロビジョニングする前にVDAの基本イメージ上でNgenを実行してください。これにより、プロビジョニング後の仮想デスクトップのバックグラウンドでNgenが長時間実行されるのを避けることができます。[#381437]
  - アップグレード中、PICAlsPorticaV2エントリポイントが見つからないという内容のメッセージが表示されることがあります。このメッセージは無視できます。アップグレード処理を完了して、指示に従ってマシンを再起動してください。[#423947]
  - 評価版のWindows 8へのVDA for Windows Desktop OSのインストールに失敗することがあります。この場合、「このオペレーティングシステムにはインストールできません」というメッセージが表示されます。この問題は、Windows 8の評価版がサポート対象外として認識されるために発生します。この問題を解決するためのHotfixがCitrix Knowledge CenterのCTX139660で提供されています。インストールメディアの内容を任意のハードディスク上にコピーして、そこにこのHotfixの内容を抽出してからWindows 8の評価版へのインストールを行ってください。詳しくは、上記Citrix Knowledge Centerのページを参照してください。
  - 製品のインストール時に、Provisioning Servicesで作成されたマシンで問題が発生することがあります。この問題は、インストール時に.NET Framework 3.5が検出されないことと発生します。この問題を避けるには、.NET FrameworkのVersion 2.0、3.0、3.5、4.0、4.5、および4.5.1がすべてインストールされていることを確認してください。[#442639, #447851]
- 
- XenDesktop 7.6には、CloudPlatformのボリュームワーカーパッケージの新しいバージョンが含まれています。既存のボリュームワーカーテンプレートをこの新しいバージョンにアップデートすることを強くお勧めします。ただし、このパッケージではインプレースアップグレードがサポートされないことに注意してください。このため、事前に既存のパッケージを完全にアンインストールするか、新しいボリュームワーカーテンプレートを最初からビルドする必要があります。[#493211]

- XenDesktop 5.6 FP1からのアップグレード後に、Remote PC Access Serviceの管理者名が正しく表示されない場合があります。これは表示上の問題であり、実際の動作には影響しません。  
[#437948]
  - オフィスにあるユーザーのマシンが休止状態になると、リモートPCアクセスセッションの起動に失敗することがあります。この問題が発生したら、セッションの起動を再試行してください。灰の再接続画面が表示されるラップトップコンピューターでは、PCを再起動する必要があります。これを行うには、管理コンソールで [強制シャットダウン] / [強制再起動] を実行するか、ローカルで再起動します (ただし未保存のデータが失われます)。  
[#441154]
  - Intel社のActive Management Technology (AMT) やWake on LANパケットの代わりにウェイクアッププロキシを使用すると、マシンの起動に失敗することがあります。これは、Microsoft System Center Configuration Manager側の問題です。  
[#441412]
  - スマートカードリーダーが接続されたユーザーデバイス上でVDAを起動し、Desktop Viewerで [切断] を選択すると、次のVDAの起動に失敗することがあります。この場合、スマートカードでのログオン画面が表示されたり、「スマートカードの読み取り中」というメッセージが表示されたりします。この問題を解決するには、次のいずれかの操作を行います。
    - スマートカードリーダーからスマートカードを取り出して再挿入します。
    - [キャンセル] をクリックします。その後、VDAでCtrl+Alt+Delキーを押します。  
[#322301]
  - 単一のWindows 2008 R2 SP1マシンにXenAppソフトウェアとXenDesktopソフトウェアをインストールした後で、StoreFrontとStudioを同時に起動しようとすると、Citrixコンソールが異常終了します。この問題は、インストールの最後にショートカットメニューでStoreFrontとStudioを同時に起動したり、コンソールでStoreFrontを開いてからStudioを開いたりすると発生します。この問題を解決するには、Native Image Generator (Ngen) を使用して.Netのネイティブイメージを強制的にアップデートします。これを行うには、コマンドプロンプトでc:\windows\microsoft.net\framework64\に移動して、コマンドngenupdate /forceを実行します。この処理には数分かかる場合があります。  
注：framework64のバージョン番号が異なる場合がありますが、常に「2.0」で始まる必要があります。  
[#490819]
  - Directorでライセンスエラーが正しく表示されず、Studioでライセンスサーバーと通信できなくなることがあります。この問題は、同じXenDesktop/XenAppサイトを管理する複数のStudioインスタンスを開いたままデリバリーサイトのライセンスサーバーアドレスを変更すると発生します。この問題を解決するには、すべてのStudioインスタンスを閉じてから単一のインスタンスを開き、[ライセンス] ノードを開きます。これにより、ライセンスデータが更新されます。  
[#492971]
  - VMware vSphereでMachine Creation Services (MCS) を使用してマシンをプロビジョニングする場合、プロビジョニングされたマシンのCPUソケットとCPUコアは、カタログ作成時に使用された基本イメージのものと同じになります。MCSカタログの作成時に仮想CPUの数をホストでの最大数よりも大きくすると、自動的にホストでのCPUソケットとCPUコアの最大数に基づく仮想マシンプロビジョニングされます。このとき、メッセージは表示されません。  
[#331269]
  - ビュアIPV6環境でSystem Center Virtual Machine Managerを使用し、Machine Creation Servicesでマシンカタログを作成する場合、マスター仮想マシンでIPV4が構成されていなくても、すべての仮想マシンにIPV4およびIPV6の両方が構成されます。これはMicrosoft製品の問題であり、現在回避策はありません。  
[#371712]
  - IPV4とIPV6が使用されるVMware vSphere環境でVMware ESXハイパーバイザーにVMXNET3ネットワークアダプターが構成されている場合、マスター仮想マシンでIPV4が構成されていなくても、すべての仮想マシンにIPV4およびIPV6の両方が構成されます。これはVMware製品の問題であり、現在回避策はありません。  
[#371712]
  - マシンカタログ名およびストレージパス名に長い文字列を使用すると、VMMジョブウィンドウでディスクの接続エラーが発生することがあります。Microsoft社によると、仮想マシンリソースのファイルパスは255文字以下である必要があります。この問題は、スタンドアロンのHyper-Vホストでローカルストレージを使用する場合に、仮想マシンの格納先のパス名が長いと発生することが確認されています。ただし、この問題はほかのHyper-Vホストでも発生する可能性があります。  
[#359673]
- この問題を解決するには、以下の手順に従います。
- 特にストレージのパスが長い場合は、VMM MCSカタログを作成するときに短い名前を指定してください。
  - 仮想マシンの格納先ストレージのパスを短くしてください。
  - 以前使用していたデータベースに切り替える場合、SDKを使用する必要があります。Studioは使用できません。  
[#355993]
- 構成ログデータベースを切り替える場合：
- Set-LogDbConnection -DataStore 'Logging' -DBConnection \$null
  - Set-LogDbConnection -DataStore 'Logging' -DBConnection 'new database connection string'
- 監視セカンダリデータベースを切り替える場合：
- Set-MonitorDbConnection -DataStore 'Monitor' -DBConnection \$null
  - Set-MonitorDbConnection -DataStore 'Monitor' -DBConnection 'new database connection string'
- ここで、<new database connection string>は以下のようなデータベース接続文字列です。
- ```
'Server=dbserver;Initial Catalog=dbname;Integrated Security = True'
```
- ハイパーバイザー上でMachine Creation Servicesを使用してマシンをプロビジョニングする場合、CPUは追加されますがCPUコアが追加されません。カタログの作成時にCPU値を変更すると、CPI単位の製品ライセンスを追加しなければならなくなる場合があります。たとえば、マスターイメージのCPUが1つでコア数が4つの場合に、カタログを作成するときにCPU値を変更します。この場合、MCSカタログの作成時に仮想CPUの数をホストでの最大数よりも大きくすると、プロビジョニングされた仮想マシンにはホストでサポートされる最大数のCPUが割り当てられます。このとき、メッセージは表示されません。たとえば、デスクトップOSマシンでは2つの物理CPUしか使用できないため、3つ以上を割り当てても2つしか割り当てられません。
- この問題を回避するには、マスターイメージ仮想マシンの仮想CPUの数をカタログで展開する仮想マシンと同じにしてください。  
[#331274]
- デスクトップOSマシンまたはサーバーOSマシンからLync 2013クライアントを配信する場合、ビデオチャット機能を使用することはできません。Lync 2013を使用する場合は、「XenDesktop 7、XenApp 6x、およびCitrix Receiver 4.0でのMicrosoft Lync 2013 VDIプラグインのサポート」 (<http://support.citrix.com/article/CTX139561>) を参照してください。  
[#371818]
  - デスクトップエクスペリエンス拡張を有効にしても、Windowsサーバー上のシームレスウィンドウモードのアプリケーションウィンドウにAeroテーマが適用されない場合があります。
  - シームレスアプリケーションのみを起動するユーザーにはAeroテーマが適用されません。
  - シームレスアプリケーションとデスクトップを組み合わせて起動するユーザーには、最初のデスクトップセッションを確立した後にAeroテーマが適用され、その後でシームレスアプリケーションにAeroテーマが適用されます。
- この問題を解決するには、デフォルトのユーザープロファイルでCitrix Enhanced Desktopテーマを有効にします。これにより、VDAのすべてのユーザーのシームレスアプリケーションウィンドウにデスクトップエクスペリエンス拡張が適用されます。このテーマはVDAと一緒にインストールされ、すべてのVDAでこの設定を行う必要があります。  
[#348812]
- Windows 8マシンにStudioからメッセージを送信すると、ホーム画面ではなくデスクトップ画面に表示されます。メッセージを表示するにはデスクトップモードに切り替える必要があります。これはサードパーティ製品の問題であり、現在回避策はありません。  
[#387356]
  - サイト内のハイパーバイザーインフラストラクチャと同じ名前前の複数のオブジェクトがある場合、Studioに正しく表示されません。ハイパーバイザーインフラストラクチャでネットワークやスレーズなどのオブジェクトを作成するときに、一意の名前を指定してください。  
[#384959]
  - 管理権限の委任機能を使って新しいスコープを持つ管理者を作成した後で、Studioの表示を更新する必要があります。これを行わないと、新しい管理者がログオンして接続リソースを作成するときに権限エラーが発生することがあります。  
[#386634]
  - ランダムに割り当てられるデスクトップOSマシン上のアプリケーションが起動に失敗することがあります。これは、アプリケーションの起動に時間がかかってタイムアウト (デフォルトで1分) が発生し、セッションが自動的に終了するためです。  
[#389025]
- この問題を解決するには、管理者が基本イメージを変更して、次のようにタイムアウト設定を変更してから再度プロビジョニングしてください。

レジストリキーHKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshell\TWIを開きます (ない場合は作成します)。



値の名前: ApplicationLaunchWaitTimeoutMS

種類: REG\_DWORD

値のデータ: <追加のタイムアウト時間 (ミリ秒) >

注: 設定可能な最小値は10秒であるため、10000よりも小さい値を設定すると10000に置き換えられます。

- デリバリーグループでアプリケーションを検出するには、登録済みで電源をオンにできるマシンがいくつか存在している必要があります。App-Vアプリケーションの場合は、StudioでApp-V公開構成する必要があります。[#393676]
- ユーザーが最初のApp-Vアプリケーションを起動した直後に2つ目のApp-Vアプリケーションを起動しようとすると、起動に失敗することがあります。この問題を回避するには、初回同期が完了するのを数分待ってから2つ目のアプリケーションを起動する必要があります。この問題について詳しくは、[CTX139556](#)を参照してください。[#397521]
- アンチウイルスプログラムBitDefenderをVDA上にインストールすると、マシンカタログを作成できないことがあります。現在、この問題を回避する方法はありません。[#392705]
- XenDesktop 7.5へのアップグレード後にCloudPlatformまたはAmazon Web Services (AWS) への接続を作成できなくなることがあります。この問題は、そのサイトでオンプレミスのハイパーバイザーへの接続が構成されていると発生します。この問題を回避するには、新しいXenDesktop 7.5サイトを作成してください。このようなハイブリッド展開環境 (クラウドと非クラウドのサイトが混在する環境) では、同じStoreFrontサイトを共有するサイトを別々に作成してください。[#454114]
- VMM 2012 SP1とHyper-V 2012の環境でODXが有効なSANストレージを使用すると、マシンカタログの作成に失敗します。この問題の解決方法については、<http://support.citrix.com/article/CTX139333>を参照してください。[#424040]
- Studioのポリシー編集ウィンドウには、Citrixポリシー設定の**CAリスナー接続タイムアウト**が7.0 Desktop OS、7.1 Desktop OS、7.5 Desktop OS、および7.6 Desktop OSに適用されると表示されますが、この設定はVirtual Delivery Agent 5.0、5.5、および5.6 Feature Pack 1にのみ適用されます。
- Studioに表示されるCitrixポリシー設定の**クライアント接続での非公開アプリケーションの起動**の説明が、サーバー上でICAまたはRDPを介して開始アプリケーションを起動するか、公開アプリケーションを起動するかを指定するという間違っただけの内容になっています。この設定は、サーバー上でRDPを介して開始アプリケーションを起動するの、公開アプリケーションを起動するのだけを指定します。
- タグによってフィルターされると、Citrixポリシーが正しく適用されないことがあります。これを防ぐには、[CTX142439](#)からHotfixを入手して適用します。[#529165]

- ホストされるアプリケーションの使用量レポートで、時間範囲が過去7日間を超える場合、ユーザーごとのアプリケーション使用量データが正しく表示されません。この問題は、過去7日間を超えるアプリケーションデータが削除されるために発生します。ユーザーごとのアプリケーション使用量データを正しく表示するには、過去7日間にレポートされたデータのみを使用してください。[504642]
- Directorの [傾向] ページと [フィルター] ページにポップアップエラー「Unexpected error. Check your network connection or view server event log for further information (予期しないエラーです。ネットワーク接続をチェックするか、詳細についてサーバーのイベントログを参照してください)」が表示されることがあります。このエラーは、XenDesktopデータベースの名前にスペースが含まれていると発生します。Directorで使用されるAPIコールでは、名前にスペースが含まれているデータベースがサポートされません。データベース名の要件に従って適切なデータベース名を指定してください。[494339]
- アップグレードにより、Directorで管理者のUI.GlobalSearchResults設定が失われるという問題があります。この問題を解決するには、アップグレード後Iweb.configファイルのUI.GlobalSearchResults設定を手作業で編集してください。[484066]
- デスクトップ数が20,000を超える環境に多くのHotfixを同時に適用すると、Hotfixインベントリの更新に数時間かかることがあります。[#489604]
- 既存のデリバリーグループを削除して同じマシンカタログを使ってデリバリーグループを再作成すると、履歴情報が保持されます。このデータにより、[傾向] ページの表およびグラフに不適切な履歴情報が表示されます。傾向履歴データを保持せずにデリバリーグループを作成するには、マシンカタログを削除して新しいマシンカタログを作成してからデリバリーグループを再作成してください。[#480010]
- [フィルター] ページから [ユーザーデバイス] ページに移動すると、ユーザー名が表示されなくなることがあります。Active Directoryでの制限によりDelivery Controllerでドメインにアクセスできない場合に、ユーザーのフルネームが [ユーザーデバイス] ページに表示されなくなることがあります。この問題を回避するには、XenDesktopがインストールされているドメインがエンドユーザーのドメインにより信頼されていることを確認してください。[#479517]
- 既存のサイトグループ内のすべてのサイトに委任管理者が割り当てられていない場合、そのサイトグループのマシンおよびエンドポイントデバイスの検索に失敗します。管理者がサイトグループ内のすべてのサイトに検索を実行できるようにするには、それらのサイトの管理権限を持つ委任管理者グループにその管理者を追加してください。[#491740]
- Windows XPの仮想デスクトップ上でWinRM 2.0が実行され、VDAのバージョンが7.0よりも古い (つまりVirtual Desktop Agentの) 場合、WinRMポートのリスニング順を変更しないとDirectorで監視できません。設定を5985.80に変更する方法については、「[詳細構成](#)」を参照してください。[#273609]
- Director Configuration Toolを使用して既存のサイトグループを編集することはできません。サイトグループを編集するには、Director ConfigurationフォルダーのXML構成ファイルを開いてください。[#491681]
- Active Directoryユーザーおよびコンピューターでログオンスクリプトによるユーザーの構成に失敗し、[ログオン期間] パネルにデータが表示されません。ログオンスクリプトでユーザーを構成するには、グループポリシーを使用する必要があります。[#393259]
- Personal vDisk VDAに対する初回ログオンからのログオン処理時間データが、Directorで収集されないまたは表示されないことがあります。2回目以降のログオンではこの問題は発生しません。[#383941]
- Windows Server 2012デスクトップからコンソールセッションを実行する場合、Directorの [ユーザーの詳細] ページを開くとエラーが発生してデータが表示されません。

回避策としてVDAを登録し、現在のセッションからログオフして、再度VDAにログインします。[#388513]

- Directorの [ダッシュボード] ページの [インフラストラクチャ] パネルで、Citrix CloudPlatformベースのホスト接続、Amazon Web Services、Hyper-V、およびMicrosoft System Center Configuration Managerの状態で「使用不可」として表示され、正しい情報が表示されません。[#449806, #446397]
- Directorの [ダッシュボード] ビューでは、ホストされるアプリケーションを起動した場合のログオン期間の情報は更新されません。[#386860]

注: HDX Flash互換に対する最新の更新については、[CTX136588](#)を参照します。

- 新しいマシン上のHDX MediaStreamリダイレクトの構成に失敗することがあります。この問題を解決するには、Receiverをインストールした後でエンドポイントマシンを再起動してください。[#494741]
- SHA512アルゴリズムを使用した証明書を使用すると、セッションへのSSL/TLS接続に失敗します。この問題は、Windows Server 2012 R2が動作するサーバーOSマシンのセッションをReceiver for HTML5を使用して起動すると発生します。この問題を避けるには、このようなシナリオでSHA512アルゴリズムの証明書を使用しないでください。[#487284]
- GPUバススルーおよびNVIDIA Keplerカードを使用する場合、3つまたは4つのモニターが構成されたHDX3D Proユーザーデバイスで最初の接続に失敗することがあります。この問題が発生した場合は、再接続してください。[#422049]
- ユーザーが接続を試行するときに画面解像度やセッション画面のサイズを変更すると、画面がちらついたり黒く表示されたりすることがあります。この問題は、高い解像度が選択され、起動時にビデオドライバーで十分なメモリを割り当てられない場合に発生します。現在のメモリ設定を確認するには、[画面の解像度] ダイアログボックスの [詳細設定] を選択します。高い解像度をサポートするメモリが割り当てられるようにするには、VDAのメモリを増やしてください。一般的に次の式で必要なメモリ量を計算できます。Windows 7: モニター数× (幅×高さ×4バイト/ピクセル×2バックバッファ)。Windows 8: モニター数× (幅×高さ×4バイト/ピクセル×3バックバッファ)。[#494671]
- USBリダイレクトポリシーが有効な場合に、ユーザーがUSBストレージやオーディオデバイスのリダイレクトを選択しても正しくリダイレクトされません。この問題は、Windows Server 2012以降が動作するサーバーOSマシンでのみ発生します。これらのデバイスをリダイレクトから除外するには、ポリシーの [クライアントUSBデバイスリダイレクト規則] 設定を使用します。USBリダイレクト規則およびその構成について詳しくは、<http://support.citrix.com/article/CTX137939>を参照してください。[#479578]

- 管理者がセッション画面の保持ポリシーの設定を無効から有効に変更すると、サーバーOSマシンからセッションを開始するときにReceiverに「<アプリケーションまたはデスクトップの名前>へ接続に失敗しました。状態 (1030)」という内容のエラーメッセージが表示されることがあります。この問題を解決するには、管理者がVDAを再起動してポリシーの変更内容を適用する必要があります。[#486073]
- リモートPCアクセスでLync 2013のセッションをリモートで実行するときに、クライアントデバイス側のWebカメラがLync 2013のカメラの一覧に表示されない場合があります。この問題を解決するには、ユーザーがリモートマシン上のデバイスマネージャーでWebカメラを無効にする必要があります。[#482807]
- このリリースではハードウェア圧縮がサポートされないため、WebカメラがH.264圧縮をサポートしている場合でもソフトウェア圧縮を使用する必要があります。これを行うには、ユーザーデバイス上のレジストリHKEY\_CURRENT\_USER\Software\Citrix\HdxRealTimeにDWORDレジストリ名DeepCompress\_ForceSWEncodeを追加します。この値を1に設定すると、ソフトウェア圧縮が使用されます。ソフトウェア圧縮は、デフォルトでは無効になっています。[#357356]
- ビデオ解像度を720p (1280×720) よりも高くすると、HDX RealTime Webカメラビデオで遅延が発生します。[#350187]
- HDX Flashリダイレクトを継続的に使用すると、セッションが応答不能になることがあります。[#350085, #361926]
- HDX RealTime WebカメラではWebカメラでサポートされるほとんどのRaw形式がサポートされますが、まれにサポートされない場合があります。この機能が正しく動作しません。[#338318]
- HDX Webカメラを使用する場合、一部のWebカメラで画像が重複して表示されることがあります。[#367322]
- HDX RealTime Webカメラは、以下のアプリケーションではサポートされていません。
  - Windows Server 2012を実行するサーバーOSマシンでホストされたCitrix GoToMeeting。[#346430]
  - GoToMeetingでオンライン会議の開始後にWebカメラを装着した場合。[#346140]
  - Windows 8、Windows 8.1、またはWindows Server 2012上のVDAを使用するMicrosoft Lync 2013およびAdobe Connect。[#340784, #348506, #459732]
  - Microsoft Office Communications Server (OCS) のビデオ通話で、通話中にWebカメラを装着した場合。[#370236]
  - Microsoft Silverlight。この問題は断続的に発生します。これを行うには、ユーザーデバイス上のレジストリHKEY\_CURRENT\_USER\Software\Citrix\HdxRealTimeにDWORDレジストリ名DeepCompress\_ForceSWEncodeを追加します。[#379779]
  - 64ビットのビデオ会議アプリケーション。64ビットアプリケーションでのビデオ圧縮はサポートされていません。[#366515]
- Windows XP仮想デスクトップ上のXenDesktop 5.6 Feature Pack 1 Virtual Desktop AgentでHDX 3D Proを使用する場合、初回接続時に [微線画] チェックボックスがオンになり、灰色表示になることがあります。この問題は、レジストリ更新の遅延により、構成ツールのユーザーインターフェイスが初期化時に不正なデフォルト値を使用するために発生します。[#353031]  
この問題を解決するには、セッションを切断してから再接続します。

問題が解決されない場合は、前回のセッション情報を消去します。これを行うには、以下のレジストリ配下の設定を削除します。

HKEY\_CURRENT\_USER\Software\Citrix\HDX3D\BitmapRemotingConfig

- Windows XPで、Citrix HDX Webカメラが正しく検出されない場合があります。この問題を回避するには、<http://www.microsoft.com/ja-jp/download/details.aspx?id=14431>からMicrosoft Visual C++ 2005 Service Pack 1再頒布可能パッケージをインストールして再試行してください。[#382733]
- デバイスマネージャーの [ディスプレイアダプター] ノードで、 [標準VGAグラフィックアダプター] に黄色の警告アイコンが表示されることがあります。ただし、機能に問題は生じたいためこのアイコンは無視して構いません。このアイコンは、新しいモデルのXPDMディスプレイドライバ (Citrix Display Driver) がロードされている場合に従来のモデルのXPDMディスプレイドライバ (標準VGAグラフィックアダプター) のロードが許可されないために表示されます。[#339390]
- Windows 8ユーザーデバイスで、メディアファイルの再生時に問題が発生することがあります。この問題は、ユーザーデバイスにメディアファイルをストリーム配信するときのクライアント側のコンテンツ取得プロトコル用のデフォルトプログラムを、正しく登録できないために発生します。Microsoft Media Streaming (MMS) およびReal Time Streaming (RTS) プロトコルでこの問題を回避するには、メディアファイル用のデフォルトプログラムをWindows Media PlayerからCitrix CSF Handlerに変更してください。Hypertext Transfer Protocol (HTTP) プロトコルの問題に対する回避策はありません。[#328805]
- ホストされる共有デスクトップおよびアプリケーションでTWAINリダイレクトに失敗することがあります。これは、TWAINバイナリが特定のパス上にない動作しないサードパーティ製TWAINアプリケーションの問題です。[#300854, #340999]  
この問題を回避するには、VDAを実行するWindows Server 2012マシンで、以下のファイルを特定の場所にコピーしてください。
  - twain\_32.dllをユーザープロファイルのWINDOWSフォルダー (%USERPROFILE%\Windows\) にコピーします。
  - twain\_32.dllmuiをユーザープロファイルのWINDOWS\en-USフォルダー (%USERPROFILE%\Windows\en-US\) にコピーします。
- HDX MediaStream Windows Mediaメディアリダイレクトが無効な環境では、64ビットのWindows Media PlayerまたはQuickTime Playerで一部のビデオファイルをサーバー側でレンダリングできません。この問題を回避するには、32ビットのWindows Media Playerを使用してください。[#384759]
- 仮想デスクトップで選択されたUniversal Print ServerプリンターがWindowsコントロールパネルの [デバイスとプリンター] に表示されない場合があります。この問題が発生しても、アプリケーションからこのプリンターを使って正しく印刷できます。この問題は、Windows Server 2012、Windows 10、およびWindows 8プラットフォームでのみ発生します。[#335153]
- GPUバススルーおよびNVIDIA Keplerカードを使用する場合、3つまたは4つのモニターが構成されたHDX3D Proユーザーデバイスで最初の接続に失敗することがあります。ただし、2回目以降の接続には成功します。[#422049]
- USBリダイレクト用に構成されたWebカメラを使ったGoToMeetingセッションがIntel Core i7プロセッサベースのコンピューターのリモートPCアクセスセッションで開始されると、ユーザーのWindowsコンピューターが応答を停止します。この場合は、ユーザーのコンピューターを再起動して、リモートPCアクセスセッションを再開してください。セッションは、切断されたときの状態から再開されます。この問題の発生を避けるには、USBリダイレクトの代わりにHDX Webカメラビデオ圧縮を使用してください。詳しくは、<http://support.citrix.com/proddocs/topic/xendesktop7/hd-new-graphics-video.html>を参照してください。[#423284]
- ユーザーデバイスでReceiver for HTML5を実行する場合、Windows Server 2012 R2が動作するサーバーOSマシンに接続できないことがあります。この問題を解決するには、以下の操作を行ってください。
  - Machine Creation Services (MCS) またはProvisioning Servicesで作成されたマシンではなく、既存のマシンをWindowsサーバーOSマシンとして使用します。
  - MCSで作成されたマシンを使用する場合は、使用するマスターイメージ上でレジストリを編集して以下のDWORD値を作成します。  
注意：レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があります。Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。  
HKEY\_LOCAL\_MACHINE\Software\Citrix\GroupPolicy\Defaults\lcaPolicies\AcceptWebSocketsConnections = 1 HKEY\_LOCAL\_MACHINE\Software\Citrix\GroupPolicy\Defaults\lcaPolicies\AllowCWebSocketのポート番号はレジストリで設定されるため、Websockets接続のCitrixポリシーをカタログに適用する必要はありません。
- Provisioning Servicesで作成されたマシンを使用する場合は、[CTX139846](#)の推奨事項に従ってください。[#424064]
- デュアルモニターセットアップでHDX Flashリダイレクトが使用され、1つのモニターでFlashコンテンツウィンドウが全画面モードの時、画面上の任意の場所をクリックするとFlashコンテンツウィンドウからフォーカスが失われてセッションウィンドウの後ろに隠れてしまうことがあります。これはHDX Flashリダイレクトの仕様です。この現象が起こった場合、Citrix Desktop Viewer ツールを使ってセッションウィンドウを全画面モードからウィンドウモードに変更すると、Flashコンテンツウィンドウを再度表示できます。[#567132]

- 最新バージョンのライセンスサーバーがインストールされていない場合、Delivery Controllerのインストール時にエラーが発生することがあります。Version 7.6にアップグレードする場合は、最初に最新バージョンのライセンスサーバーをインストールしてください。[#510425]
- ライセンスサーバーをアンインストールして再インストールした場合、読み取りのみが許可された管理者がStudioでライセンス情報を表示しようとすると、その操作を行うための権限がないというメッセージが表示されます。これは、この管理者に新しいライセンスサーバーを信頼する権限がないために発生します。この問題を回避するには、すべての管理作業を実行できるライセンス管理者が [ライセンス] ノードで新しいライセンスサーバーを認証する必要があります。[#380982]
- ライセンスサーバーをインストールしたときにライセンスサーバー設定ツールで正しく構成しない場合、それ以降のライセンスサーバーのアップグレードに失敗します。この問題を回避するに

- は、すべてのライセンスサーバーをインストール後のライセンスサーバー設定ツールで構成してください。[#377079]
- ライセンス管理コンソールやSimple License ServiceのWebページインターフェイスを開いたときに、空白のページが表示されることがあります。この問題は、Internet Explorerセキュリティ強化構成が有効になっていて、ライセンス管理コンソールやSimple License ServiceのWebページインターフェイスが信頼済みサイトに追加されていない場合に発生します。回避方法：Internet Explorerセキュリティ強化の構成を無効にしてください。[#382429]
  - 製品のインストールまたはアップグレード時にポート8083がほかのアプリケーションで使用されていると、ライセンスサーバーの構成およびインストールに失敗します。この問題を回避するには、まずイベントログをチェックしてこのポートが使用中かどうかを確認します。ほかのアプリケーションがこのポートを使用している場合は、以下の操作を実行します。
    - インストールメディアのx64\LicensingフォルダーでCTX\_Licensing.msiをダブルクリックして、ライセンスサーバーをアンインストールします。
    - 製品のインストーラーを再度実行します。インストール済みのコンポーネントがいくつか表示されます。[インストール]をクリックします。インストールが完了し、各コンポーネントが構成されます。
    - ライセンスサーバーを手動でインストールし、マシン上のほかのアプリケーションと競合しないポート番号を指定します。[#390815]
  - ライセンス管理コンソールやCitrix Simple License ServiceのWebページインターフェイスに表示されるユーザー一覧では、ユーザー名やグループ名の非ASCII文字が正しく表示されません。このため、ロシア語版のオペレーティングシステム上では組み込みの管理者グループがユーザー一覧に追加されません。この問題は、ライセンスサーバーの新規インストールおよびアップグレードインストールの両方で発生します。以前のリリースのXenDesktopおよびSimple License Serviceで組み込みの管理者グループに属していたユーザーは、アップグレード後にライセンス管理コンソールにアクセスできなくなります。この問題を解決するには、アップグレード後にライセンス管理コンソールを使用してASCII文字によるユーザー/グループ名を追加してください。または、ほかの言語版のWindowsオペレーティングシステムにライセンスサーバーをインストールしてください。[#395305]
  - 製品をインストールまたはそれにアップグレードした後で、Studioに永続的ライセンスの有効期日が01/01/2000と表示されることがあります。この期日は無視でき、デスクトップやアプリケーションを起動できます。[#402975]
  - Windows 8またはWindows Server 2012のホスト環境でローカルアプリケーションアクセスが有効な場合、クライアント側のアプリケーション用の拡張子にファイルタイプの割り当て (FTA) がいとリダイレクトに失敗します。ユーザーは [ストアでアプリを探す] または [その他のオプションの表示] を選択するよう求められます。[#372834] この問題を回避するには、VDAマスターイメージ上で以下のいずれかを行います。
    - HKEY\_CLASSES\_ROOT\Unknown\shell\openas\command\DelegateExecuteのDelegateExecuteレジストリ値の名前を変更します。
    - メモ帳を使ってその拡張子のファイルを開きます。以降は、FTAリダイレクトが実行されます。
  - Windows Server 2012上ではデフォルトでURLリダイレクトが無効になっています。これを有効にするには、Internet Explorer拡張構成モードを無効にしてください。[#356260]
  - セッション中に変更したローカルアプリケーションアクセスのプロパティは自動的に適用されません。この問題を回避するには、いったんログオフして再ログオンしてください。[#357488]
  - Windows 8またはWindows 2012プラットフォーム上でローカルアプリケーションアクセスアプリケーションを起動すると、Virtual Delivery AgentをModern Shellから起動できません。この問題を回避するには、ローカルアプリケーションを閉じてからVirtual Delivery Agentアプリケーションを起動します。[#359670]
  - ローカルアプリケーションが起動している場合、Receiver for Windows 3.4またはそれ以前でShellhook.dllがロードされないという問題があります。この問題を回避するには、HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\WindowsでLocalAppInit\_DLLsのレジストリ値を1に変更してください。この回避策を使用できない場合は、ファイルタイプの関連付けによるクライアントからサーバーへのリダイレクトが正しく機能しません。[#356130]
  - ポップアップブロックが有効になっているインターネットサイトで、URLリダイレクトに失敗することがあります。この問題を回避するには、ポップアップブロック機能を無効にします。ただし、セキュリティ上の理由により、ポップアップブロックの無効化は推奨されません。[#371220]
  - VDAセッションでAeroモードが有効な場合、VDA側のアプリケーションとクライアントデバイス側のアプリケーションで動作の相違が生じることがあります (Alt+Tabキーを押してアプリケーションを切り替えるときなど)。一貫した動作を提供するには、Aeroモードを無効にしてください。[#361043]
  - FlashリダイレクトにはWMPブラックスクリーンおよび範囲外のフラッシュ疑似コンテナーウィンドウなど、互換性の問題があります。この問題を回避するには、Flashリダイレクトを無効にしてください。[#360182]
  - ローカルアプリケーションアクセスが有効な場合、デスクトップコンポジションで高品位なグラフィックを表示する機能は無効になります。この機能を使用するには、ローカルアプリケーションアクセスを無効にしてください。[#377386]
  - セッション画面の保持とローカルアプリケーションアクセスが有効な場合に、ツールバーの [ホーム] をクリックしてクライアントのデスクトップを表示した後にネットワークが切断されると、VDAセッションの画面が保持されないことがあります。この場合、ツールバーとクライアントのデスクトップだけが表示されます。このツールバーでは、[切断] オプションだけを実行できます。[#357769]
  - ローカルアプリケーションアクセスが有効で、メディアファイルを再生時にユーザーがセッションを全画面表示にすると、イメージの一部またはすべてが表示されません。この問題を解決するには、セッションを再起動してください。[#402702]
  - ドメインに参加したWindows 8マシン上にCitrix Desktop Lockをインストールしてそれを再起動した後、デスクトップの背景が黒になり、通常のスタート画面の代わりに [マイコンピュータ] ウィンドウが表示されることがあります。マシンをログオフするには、「logoff」とウィンドウのアドレスバーに入力し、Enterキーを押します。Windows 8マシンとDesktop Lockの構成について詳しくは、「Desktop Lock」を参照してください。[#329075]
  - Windows 8またはWindows 7のマシンに障害が発生してユーザーがCtrl+Alt+Delキーを押すと、ローカルデスクトップのWindowsセキュリティ画面が表示されます。この問題は、Citrix Desktop Lockソフトウェアが使用されている場合でも発生します。この問題を回避するには、[タスクマネージャーの起動] を選択して [再起動] 画面を開いてください。[#337507]
  - Citrix Desktop Lockはドメインに参加しているユーザーデバイスにAdobe Flashコンテンツをリダイレクトしません。コンテンツの閲覧は可能ですが、これはローカルではなくサーバーで実行されます。この問題を回避するには、サーバー側でのコンテンツ取得機能によってサーバーからユーザーデバイスにコンテンツが渡されるように、Adobe Flashリダイレクトを構成してください。この問題は、ドメインに参加していないデバイス上またはDesktop Viewerでコンテンツを閲覧する場合には発生しません。[#263092]
  - Desktop LockおよびReceiver for Windows Enterprise 3.4がインストールされたユーザーデバイスでWindows 7を実行している場合、VDAを開始できないことがあります。この問題は、インターネットに接続されているかどうかに関係なく発生します。この問題が発生すると、「No Internet Connectivity」というエラーメッセージが表示されます。これはサードパーティ (Microsoft社) の問題です。詳しくは、<http://technet.microsoft.com/ja-jp/library/cc766017>を参照してください。[#408642]
  - 32ビットのWindows XPクライアントセッションでWindows Media Playerファイルを継続的に再生すると、ユーザーデバイス上でwfica32.exeによるハンドルリークが発生することがあります。[#378146]
  - ユーザーが [切断] ボタンを使用できません。この問題を回避するには、オペレーティングシステムに含まれているTSDiscon.exeユーティリティへのショートカットを設定します。これにより、セッションから切断できます。[#362937]
  - ユーザーがリモートPCのセッションを切断しようとしても、セッションが切断されない場合があります。この問題は、[スタート] メニューの [切断] を選択すると発生します。[切断] を選択した後、セッションが切断されるまでに時間がかかります。この間にユーザーがCtrl+Alt+Delキーを押すと、リモートPC上のVDAが切断されるまでに数分かかります。この間、リモートのVDAが切断されるまでユーザーデバイス上のVDAが応答不能になります。[#322301]
  - Windows 7およびWindows 8上にインストールされたVDAで、マウスポインターが2つ表示されることがあります (うち1つは動きません)。この問題は、NVIDIAドライバーにより発生します。この問題を回避するには、NVIDIA GRIDテクノロジー (旧称VGX) を無効にしてください。これを行うには、MontereyEnable.exe -disable -resetを実行してマシンを再起動します。[#307921]
  - Microsoftのデスクトップコンポジション機能を有効にすると、VDAのスケラビリティが低下します。ユーザーが最大限のスケラビリティを要する場合は、デスクトップコンポジションのリダイレクト機能を使用しないすべてのユーザーのデスクトップコンポジションを無効にしてください。これを行うには、Microsoftポリシーを使用します。[#386602]
  - ユーザーがローカルでセッションのロックを解除しようとすると、セッションが自動的に再ロックされることがあります。この問題は、リモートICAセッションの実行中にリモートPCアクセス

シンのローカルキーボードを取り外したりシャットダウンしたりすると発生します。この問題を解決するには、ユーザーがセッションをリモートで再起動し、切断してからコンソール上で再ログインする必要があります。[#382554]

- ユーザーがVersion 7よりも前のVDA (Virtual Desktop Agent) を使用する場合、Windowsリモート管理 (WinRM) を適切に構成した後もDirectorにユーザーのデータが表示されないことがあります。この問題を解決するには、WinRMサービスを再起動してください。[#392047]
- セキュリティが保護された環境で、新しいVDAのControllerへの登録に失敗することがあります。この問題は、ネットワークからのアクセスを管理者にのみ許可するローカルセキュリティポリシーが構成されたVDAをインストールするときに発生します (VDAのインストールは成功します)。この場合、ユーザーのアクセス権が正しく構成されていないという警告が表示されます。詳しくは、[CTX117248](#)を参照してください。[#336203]
- Receiver for Windowsを使用する場合に、パススルー認証でストアにログオンできないことがあります。この問題は、StoreFrontの認証サービスで認証方法としてドメインパススルーが有効になっても発生します。この問題を解決するには、Controller上でWindows PowerShellコマンドSet-BrokerSite -TrustRequestsSentToTheXmlServicePort \$Trueを実行してください。[#330775]
- Windows Server 2012 VDAに対してポリシーを使って従来のグラフィックモードが指定されている場合、ICA往復チェックはサポートされません。現在、この問題を回避する方法はありません。Windows 8 VDAは従来のグラフィックモードをサポートしないため、この問題は発生しません。[#394824]
- Windows Server 2012 R2およびそれ以前のドメインコントローラーで、ユーザーの次回ログオン時にパスワードの変更を求めるとエラーが発生します。この問題を回避するには、Microsoft社の修正プログラムKB2883201を削除してください。[#438725]
- ユーザーがRDPを使ってVDAに直接接続する場合、存在していないセッションがControllerにより報告されることがあります。Studioには、そのVDAを再起動するまでこれらのセッションが接続済みの状態として表示されます。[#385823]
- FirefoxまたはChromeを使用する場合、Citrix Storefrontでパススルー認証を有効にしてもアプリケーションの起動時に資格情報の入力が必要になります。[#441487]
- ドメインパススルー認証が構成されたCitrix Receiver for Webで、ユーザーがスマートカードを使用してデバイスにログオンした場合に公開デスクトップへの接続に失敗することがあります。この問題を解決するには、デフォルトのICAファイルをストア (例: C:\inetpub\wwwroot\Citrix\Store\App\_Data\default.ica) で編集し、次の行を[Application]セクションに追加します:

```
DisableCtrlAltDel=False  
[#452813]
```

Personal vDiskに関連する既知の問題については、「[Personal vDisk](#)」を参照してください。

Provisioning Servicesの問題については、「[Provisioning Services 7.0, 7.1, および7.6について](#)」を参照してください。

# このリリースに含まれていない機能

Jan 04, 2017

以下の機能は、このリリースでは提供されません（サポートされない機能など）。

- **タッチパネルでの操作に最適化されたデスクトップ** - この設定はWindows 10マシンでは無効になっています。詳しくは、「[モバイルデバイスでの動作のポリシー設定](#)」を参照してください。
- **128ビット未満のSecureICA暗号化** - 7.xより前のリリースでは、SecureICAにより基本レベル、40ビット、56ビット、および128ビットの暗号化でクライアント接続を保護できました。このリリースのSecureICAでは、128ビットの暗号化のみを使用できます。
- **従来の印刷機能** - 以下の印刷機能は、7.xでサポートされなくなりました。
  - 従来のクライアントプリンター名を使用する機能を含む、Dosクライアントと16ビットプリンターに対する後方互換性。
  - 強化された拡張プリンタープロパティおよびWin32FavorRetainedSettingを含む、Windows 95およびWindows NTオペレーティングシステムに接続されたプリンターのサポート。
  - 自動保持および自動復元プリンターを有効または無効にする機能
  - 自動保持および自動復元プリンターを有効または無効にするサーバーのレジストリ設定であるDefaultPmFlag。サーバー上のユーザープロファイルに保存されます。
- **Secure Gateway** - 7.xより前のリリースでは、Secure Gatewayを使用してサーバーとユーザーデバイス間の接続を保護できました。このリリースでは、外部接続をセキュリティで保護するためのオプションとして、NetScaler Gatewayを使用します。
- **ユーザーのシャドウ** - 7.xより前のリリースでは、管理者はポリシーを設定してユーザー間のシャドウを制御しました。このリリースでは、エンドユーザーのシャドウ機能がDirectorコンポーネントに統合されています。管理者は、DirectorからWindowsリモートアシスタンスを使用してユーザーのアプリケーションや仮想デスクトップをシャドウして、問題のトラブルシューティングを行います。
- **電源能力管理コンポーネント** - 7.xより前のリリースでは、電源能力管理を電源の消費を減らし、サーバーの能力を管理するために使用できました。このリリースでは、Microsoft Configuration Managerを使用してこれを行います。
- **第1世代のFlashリダイレクト** - 第2世代Flashリダイレクトをサポートしないクライアント（Receiver for Windows 3.0より前のバージョン、Receiver for Linux 11.100より前のバージョン、およびCitrix Online Plug-in 12.1）では、従来のFlashリダイレクトが機能しないためにサーバー側でのレンダリングにフォールバックされます。このリリースに含まれるVDAは、第2世代のFlashリダイレクト機能をサポートします。
- **ローカルテキストエコー** - この機能は、以前の形式のWindowsアプリケーション上で入力文字列を高速に表示するために使用されました。このリリースではHDX SuperCodecおよびグラフィックサブシステムが改善されたため、この機能は削除されています。
- **Smart Auditor** - 7.xより前のリリースでは、Smart Auditorを使用してユーザーセッションの画面上のアクティビティを記録できました。このリリースでは、このコンポーネントは使用できません。7.6 Feature Pack 1では、Session Recordingにより置き換えられます。
- **Single Sign-on** - パスワードセキュリティを提供するこの機能は、Windows 8およびWindows Server 2012環境ではサポートされません。Windows 2008 R2およびWindows 7環境ではサポートされますが、7.xリリースには含まれていません。CitrixダウンロードWebサイト (<http://citrix.com/downloads>) で見つけることができます
- **Oracleデータベースのサポート** - 7.xリリースでは、SQLServerデータベースを使用します。
- **サーバーヘルスの監視および復元 (HMR)** - 7.xより前のリリースでは、この機能を使用してサーバーファーム内のサーバーでテストを実行し、サーバーの状態を監視したり、サーバーヘルスのリスクを発見したりできました。このリリースでは、Directorのコンソールを使用して、インフラストラクチャ全体のシステム状態を一元的に管理および監視できます。
- **カスタムICAファイル** - 7.xより前のリリースでは、カスタムのICAファイルを使用して、ユーザーデバイスから特定のマシンへの直接接続が可能でした。このリリースでは、この機能はデフォルトでは無効になっていますが、ローカルグループを使用する標準的な使用においては有効にできます。また、Controllerが使用できなくなった場合に高可用性モードで使用

することもできます。

- **Management Pack for System Center Operations Manager (SCOM) 2007** - このリリースでは、SCOMを使ってサーバーファームのアクティビティを監視する管理パックはサポートされていません。
- **CNAME機能** - 7.xより前のリリースでは、CNAME機能はデフォルトで有効になっていました。このため、CNAMEレコードに依存するFQDNの再ルーティングやNetBIOS名の使用に失敗することがありました。このリリースでは、Controllerの自動更新機能によりControllerの一覧が動的に更新され、サイトのControllerが追加または削除されるとVDAに自動的に通知されます。この自動更新機能はCitrixポリシーのデフォルトで有効になっていますが、ポリシーを作成することにより無効にできます。  
または、レジストリでCNAME機能を有効にして、従来のFQDNの再ルーティングやNetBIOS名の使用を許可することもできます。詳しくは、[CTX137960](#)を参照してください。
- **簡易展開ウィザード** - 7.xより前のリリースのStudioでは、このオプションを選択してすべてが構成済みのXenDesktop展開をすばやく作成することができました。このリリースではインストールおよび構成のワークフローが簡素化されたため、簡易展開ウィザードが不要になりました。
- **自動管理用のリモートPCサービス構成ファイルPowerShellスクリプト** - リモートPCは、StudioおよびControllerに統合されました。
- **Workflow Studio** - 7.xより前のリリースでは、XenDesktopのワークフロー構成用のグラフィックインターフェイスとしてWorkflow Studioを使用しました。このリリースでは、この機能はサポートされません。
- **色数** - Studioの7.6よりも前のリリースでは、デリバリーグループユーザー設定ページのこのオプションは、デリバリーグループの色数を設定しました。バージョン7.6では、デリバリーグループの色数をNew-BrokerDesktopGroupまたはSet-BrokerDesktopGroup PowerShellコマンドレットを使って設定できます。
- **クライアント接続での非公開アプリケーションの起動** - 7.xよりも前のリリースでは、このCitrixポリシー設定は、サーバー上のICAまたはRDPを介して開始アプリケーションまたは公開アプリケーションを起動するかどうかを指定しました。7.xリリースでは、サーバー上のRDPを介して開始アプリケーションまたは公開アプリケーションを起動するかどうかのみを指定します。
- **デスクトップの起動** - 7.xよりも前のリリースでは、このCitrixポリシー設定は非管理者ユーザーがデスクトップセッションに接続できるかどうかを指定しました。7.xリリースでは、VDA上のセッションに接続するため、非管理者ユーザーがVDAマシンのDirect Access Usersグループのメンバーである必要があります。デスクトップの起動設定により、VDAのDirect Access Usersグループの非管理者ユーザーは、ICAコネクションを使ってVDAに接続できます。デスクトップの起動設定は、RDP接続には影響を与えません。VDAのDirect Access Usersグループのユーザーは、この設定が有効であるかどうかには関係なく、RDP接続を使ってVDAに接続できます。
- **COMポートマッピング** - COMポートのマッピング機能により、ユーザーデバイス上のCOMポートへのアクセスが許可または禁止されます。この機能は、以前のリリースのデフォルトでは有効になっていました。このリリースでは、COMポートマッピングがデフォルトで無効になります。詳しくは、「[レジストリを使ったCOMポートおよびLPTポートリダイレクト設定の構成](#)」を参照してください。
- **LPTポートマッピング** - LPTポートのマッピング機能により、LPTポートへのレガシアプリケーションのアクセスが制御されます。この機能は、以前のリリースのデフォルトでは有効になっていました。このリリースでは、LPTポートマッピングがデフォルトで無効になります。
- **PCMオーディオコーデック** - このリリースでは、HTML5クライアントでのみPCMオーディオコーデックがサポートされます。
- **Microsoft ActiveSyncのサポート**
- **以前のバージョンのプロキシサポート** - 以下のプロキシはサポートされなくなりました。
  - Microsoft Internet Security and Acceleration (ISA) 2006 (Windows Server 2003)。
  - Oracle iPlanet Proxy Server 4.0.14 (Windows Server 2003)。
  - Squid Proxy Server 3.1.14 (Ubuntu Linux Server 11.10)。



# システム要件

Jan 04, 2017

ここでは次のことについて説明します。

[Session Recording - XenApp 7.6 FP1、FP2、およびLTSR](#)

[Delivery Controller](#)

[データベース](#)

[Studio](#)

[Director](#)

[Virtual Delivery Agent \(VDA\) for Windows Desktop OS](#)

[Virtual Delivery Agent \(VDA\) for Windows Server OS](#)

[ホスト/仮想化リソース](#)

[Active Directoryの機能レベルのサポート](#)

[HDX - デスクトップコンポジションリダイレクト](#)

[HDX - Windows Media配信](#)

[HDX - Flashリダイレクト](#)

[HDX 3D Pro](#)

[HDX - Webカメラビデオ圧縮でのビデオ会議の要件](#)

[HDX - そのほか](#)

[Universal Print Serverの要件](#)

[その他の要件](#)

ここで説明するシステム要件は、この製品バージョンがリリースされた時点で確認済みのものです。このトピックで説明されていないシステム要件コンポーネント (StoreFront、ホストシステム、ReceiverとPlug-in、Provisioning Services) については、各コンポーネントのドキュメントを参照してください。

**重要：**インストールの前に、「[インストールの準備](#)」の内容を確認してください。

特に明記されている場合を除き、コンポーネントの必須ソフトウェア (.NETやC++パッケージなど) がインストールされていないことが検出された場合、インストーラーにより自動的にインストールされます。これらの必須ソフトウェアの一部は、製品のインストールメディアにも収録されています。

インストールメディアには複数のサードパーティ製コンポーネントが収録されています。Citrixソフトウェアを使用する前に、サードパーティからのセキュリティに関するアップデートを確認して、必要に応じてインストールしてください。

インストールに必要なディスクスペースの値は参考値であり、製品イメージ、オペレーティングシステム、およびその他のソフトウェア用の容量は含まれていません。



すべてのコアコンポーネント（ControllerとSQL Server Express、Studio、Director、StoreFront、およびライセンス）を単一のサーバーにインストールする場合、製品評価のため最低3GBのRAMが必要です。ユーザー用の環境を実行する場合は、より多くのRAMが推奨されます。性能は、ユーザー、アプリケーション、デスクトップ、およびその他のほかの要因の数など、実際の構成により異なります。

**重要：** Windows Server 2012 R2システムにXenAppをインストールしたら、Kerberos Enable Tool (XASsonKerb.exe) を使用してCitrix Kerberos認証が正しく行われるようにしてください。このツールは、インストールメディアのSupport > Tools > XASsonKerbフォルダーに収録されています。このツールを使用するには、ローカルの管理者権限が必要です。Kerberos認証が正しく行われるようにするには、サーバー上のコマンドプロンプトでxassonkerb.exe -installを実行します。その後のアップデートなどでレジストリHKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Control\LSA\OSConfigが変更された場合は、このコマンドを再度実行してください。このコマンドで使用できるオプションについての情報を表示するには、-helpパラメーターを指定してコマンドを実行します。

## Session Recording - XenApp 7.6 FP1、FP2、およびLTSR

Session Recordingには、英語版、フランス語版、ドイツ語版、日本語版、スペイン語版、および簡体字中国語版があります。相互に通信するすべてのSession Recordingコンポーネントが、同じ言語である必要があります。言語を混在させたインストールはサポートされません。

英語版のSession Recordingは、英語版、ロシア語版、繁体字中国語版、および韓国語版のオペレーティングシステムでサポートされます。フランス語版、ドイツ語版、日本語版、スペイン語版、および簡体字中国語版のSession Recordingは、各言語版のオペレーティングシステムでサポートされます。

### Session Recording Administrationコンポーネント

Session Recording Administrationコンポーネント（Session Recordingデータベース、Session Recording Server、Session Recording Policy Console）は、同一サーバー上にインストールしたり、個別のサーバー上にインストールしたりできます。

### Session Recordingデータベース

以下のオペレーティングシステムがサポートされています。

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Service Pack 1を適用したMicrosoft Windows Server 2008 R2

要件：

- Microsoft SQL Server 2008 R2 SP3共有管理オブジェクトおよびMicrosoft SQL Server 2008 R2 SP3システムCLR型パッケージ。パッケージは<http://www.microsoft.com/ja-jp/download/details.aspx?id=44272>からダウンロードできます。
- Microsoft SQL Server 2014（Enterprise EditionおよびExpress Edition）、Service Pack 2を適用したMicrosoft SQL Server 2012（Enterprise EditionおよびExpress Edition）、またはService Pack 3を適用したMicrosoft SQL Server 2008 R2（Enterprise EditionおよびExpress Edition）
- .NET Framework Version 3.5 Service Pack 1

- .NET Framework Version 3.5 Service Pack 1

## Session Recording Server

以下のオペレーティングシステムがサポートされています。

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012、Service Pack 1を適用したMicrosoft Windows Server 2008 R2

要件：

Session Recordingのインストールを開始する前に、いくつかの前提条件をインストールする必要があります。サーバーマネージャーを開いてIIS役割を追加します。次のオプションを選択します。

- [アプリケーション開発] > [ASP.NET 3.5] (ほかのコンポーネントが自動的に選択されています。 [追加] をクリックして必要な役割を指定します)
- [セキュリティ]、[Windows認証]
- [管理ツール]、[IIS 6管理互換]
  - IIS 6メタベース互換
  - IIS 6 WMI互換
  - IIS 6スクリプトツール
  - IIS 6管理コンソール
- NET Framework Version 3.5 Service Pack 1
- Session Recording Serverで、通信プロトコルとしてHTTPS、および有効な証明書を使用する場合。Session Recordingでは、デフォルトでCitrixの推奨プロトコルであるHTTPSが使用されます。
- Active Directory統合を無効にし、MSMQ HTTPサポートを有効にしたMicrosoft Message Queuing (MSMQ)。

## Session Recording Policy Console

以下のオペレーティングシステムがサポートされています。

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Service Pack 1を適用したMicrosoft Windows Server 2008 R2

要件：

- .NET Framework Version 3.5 Service Pack 1

## Session Recording Agent

Session Recording Agentは、セッションを録画するXenAppおよびXenDesktopサーバーごとにインストールします。

以下のオペレーティングシステムがサポートされています。

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Service Pack 1を適用したMicrosoft Windows Server 2008 R2

要件：

- Active Directory統合を無効にし、MSMQ HTTPサポートを有効にしたMicrosoft Message Queuing (MSMQ)
- .NET Framework Version 3.5 Service Pack 1

## Session Recording Player

以下のオペレーティングシステムがサポートされています。

- Microsoft Windows 8.1
- Microsoft Windows 8
- Service Pack 1を適用したMicrosoft Windows 7

最適なパフォーマンスを得るには、Session Recording Playerを以下の条件のワークステーションにインストールします。

- 1024 X 768の画面解像度
- 32ビット以上の色数
- メモリ：1GB以上のRAM。グラフィックが多用されている録画を再生する場合、特に録画にアニメーションが多く含まれる場合は、RAMおよびCPU/GPUリソースを追加すると、パフォーマンスが向上します。

シークの応答速度は、録画のサイズやマシンのハードウェア仕様によって異なります。

要件：

- .NET Framework Version 3.5 Service Pack 1

## Session Recording Administrationコンポーネント

Session Recording Administrationコンポーネント (Session Recordingデータベース、Session Recording Server、Session Recording Policy Console) は、同一サーバー上にインストールしたり、個別のサーバー上にインストールしたりできます。

## Session Recordingデータベース

以下のオペレーティングシステムがサポートされています。

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Service Pack 1を適用したMicrosoft Windows Server 2008 R2

要件：

- .NET Framework Version 3.5 Service Pack 1 (Windows Server 2008 R2のみ)、もしくは.NET Framework Version 4.5.1または4.6。

## Session Recording Server

以下のオペレーティングシステムがサポートされています。

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012、Service Pack 1を適用したMicrosoft Windows Server 2008 R2

要件：

Session Recordingのインストールを開始する前に、いくつかの前提条件をインストールする必要があります。サーバーマネージャーを開いてIIS役割を追加します。次のオプションを選択します。

- アプリケーション開発：
  - [ASP.NET 4.5 on Server 2012 and Server 2012 R2, ASP.NET on Server 2008 R2]（ほかのコンポーネントが自動的に選択されています。[追加]をクリックして必要な役割を指定します）
- [セキュリティ]、[Windows認証]
- [管理ツール]、[IIS 6管理互換]
  - IIS 6メタベース互換
  - IIS 6 WMI互換
  - IIS 6スクリプトツール
  - IIS 6管理コンソール
- .NET Framework Version 3.5 Service Pack 1 (Windows Server 2008 R2のみ)、もしくは.NET Framework Version 4.5.1または4.6。
- Session Recording Serverで、通信プロトコルとしてHTTPS、および有効な証明書を使用する場合。Session Recordingでは、デフォルトでCitrixの推奨プロトコルであるHTTPSが使用されます。
- Active Directory統合を無効にし、MSMQ HTTPサポートを有効にしたMicrosoft Message Queuing (MSMQ)。

## Session Recording Policy Console

以下のオペレーティングシステムがサポートされています。

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Service Pack 1を適用したMicrosoft Windows Server 2008 R2

要件：

- .NET Framework Version 3.5 Service Pack 1 (Windows Server 2008 R2のみ)、もしくは.NET Framework Version 4.5.1または4.6。

## Session Recording Agent

Session Recording Agentは、セッションを録画するXenAppおよびXenDesktopサーバーごとにインストールします。

サポートされるWindowsオペレーティングシステム：

- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Service Pack 1を適用したMicrosoft Windows Server 2008 R2

要件：

- Active Directory統合を無効にし、MSMQ HTTPサポートを有効にしたMicrosoft Message Queuing (MSMQ)
- .NET Framework Version 3.5 Service Pack 1 (Windows Server 2008 R2のみ)、もしくは.NET Framework Version 4.5.1または4.6。

## Session Recording Player

以下のオペレーティングシステムがサポートされています。

- Microsoft Windows 8.1
- Microsoft Windows 8
- Service Pack 1を適用したMicrosoft Windows 7

最適なパフォーマンスを得るには、Session Recording Playerを以下の条件のワークステーションにインストールします。

- 1024 X 768の画面解像度
- 32ビット以上の色数
- メモリ：1GB以上のRAM。グラフィックが多用されている録画を再生する場合、特に録画にアニメーションが多く含まれる場合は、RAMおよびCPU/GPUリソースを追加すると、パフォーマンスが向上します。

シークの応答速度は、録画のサイズやマシンのハードウェア仕様によって異なります。

要件：

- .NET Framework Version 3.5 Service Pack 1または.NET Framework Version 4.5.1または4.6

以下のオペレーティングシステムがサポートされています。

- Windows Server 2012 R2、Standard、およびDatacenterエディション。
- Windows Server 2012、Standard、およびDatacenterエディション。
- Windows Server 2008 R2 SP 1、Standard、Enterprise、およびDatacenterエディション。

要件：

- 100MBの空きディスク容量。さらに、接続リース（デフォルトで有効）での要件を考慮する必要があります。必要なディスク容量は、ユーザー数、アプリケーション数、およびモード（RDSまたはVDI）によって異なります。たとえば、最近使用された100個のアプリケーションに対する接続リース機能を100,000人のRDSユーザーに対して有効にする場合は、約3GBの空きディスク容量が必要です。サポートするアプリケーションが多くなるほど、大きな空き領域が必要になります。専用のVDIデスクトップでは、40,000台のデスクトップに対して少なくとも400~500MBが必要です。いずれの場合も、数GBの追加領域を指定することを推奨します。
- Microsoft .NET Framework 3.5.1（Windows Server 2008 R2のみ）。
- Microsoft .NET Framework 4.5.1（4.5.2および4.6もサポートされます）。
- Microsoft .NET Framework 4.6.1。
- Windows PowerShell 2.0（Windows Server 2008 R2に付属）または3.0（Windows Server 2012 R2およびWindows Server 2012に付属）。
- Visual C++ 2005、2008 SP1、および2010再頒布可能パッケージ。

サイト構成データベース（構成ログデータベースおよび監視データベースを含む）では以下のバージョンのMicrosoft SQL Serverのバージョンがサポートされています。

- SQL Server 2016のExpress、Standard、およびEnterprise Edition。
- SQL Server 2014のSP2、Express、Standard、およびEnterprise Edition。
- SQL Server 2012のSP3、Express、Standard、およびEnterprise Edition。デフォルトでは、Controllerのインストール時に適切なバージョンのSQL Serverが検出されない場合、SQL Server 2012 SP1 Expressがインストールされます。
- SQL Server 2008 R2 SP2およびSP3のExpress、Standard、Enterprise、およびDatacenter Edition。

以下のデータベース機能がサポートされます（スタンドアロンモードのみをサポートするSQL Server Expressを除く）。

- SQL Serverクラスター化インスタンス
- SQL Serverミラーリング
- SQL Server 2012 AlwaysOn可用性グループ

ControllerとSQL Serverデータベース間の接続にはWindows認証が必要です。

サポートされるバージョンについては、[CTX114655](#)を参照してください。

以下のオペレーティングシステムがサポートされています。

- Windows 8.1、Professional、およびEnterpriseエディション。
- Windows 8、Professional、およびEnterpriseエディション。
- Windows 7 Professional、Enterprise、およびUltimateエディション。
- Windows Server 2012 R2、Standard、およびDatacenterエディション。
- Windows Server 2012、Standard、およびDatacenterエディション。
- Windows Server 2008 R2 SP 1、Standard、Enterprise、およびDatacenterエディション。

要件：

- 75 MBの空きディスク容量。
- Microsoft .NET Framework 4.6.1。
- Microsoft .NET Framework 4.5.1 (4.5.2および4.6もサポートされます)。
- Microsoft .NET Framework 3.5 SP1 (Windows Server 2008 R2およびWindows 7の場合のみ)。
- Microsoft管理コンソール3.0 (サポートされているすべてのオペレーティングシステムに付属)。
- Windows PowerShell 2.0 (Windows 7およびWindows Server 2008 R2に付属) または3.0 (Windows 8.1、Windows 8、Windows Server 2012 R2、およびWindows Server 2012に付属)。

以下のオペレーティングシステムがサポートされています。

- Windows Server 2012 R2、Standard、およびDatacenterエディション。
- Windows Server 2012、Standard、およびDatacenterエディション。
- Windows Server 2008 R2 SP 1、Standard、Enterprise、およびDatacenterエディション。

要件：

- 50MBの空きディスク容量。
- Microsoft .NET Framework 4.5.1 (4.5.2および4.6もサポートされます)。
- Microsoft .NET Framework 3.5 SP1 (Windows Server 2008 R2の場合のみ)。
- Microsoftインターネットインフォメーションサービス (IIS) 7.0およびASP.NET 2.0. IISと一緒に [静的コンテンツ] の役割サービスがインストールされていることを確認してください。これらがインストールされていない場合は、Windows Serverのインストールメディアを指定するためのメッセージが表示され、自動的にインストールされます。
- Directorを表示するための以下のWebブラウザ。
  - Internet Explorer 11および10。  
Internet Explorerの互換モードはサポートされていません。Directorを表示するには、Webブラウザの推奨設定を使用する必要があります。Internet Explorerをインストールするときに、セキュリティおよび互換性に関するデフォルトの推奨設定を適用してください。インストール済みのInternet Explorerで推奨設定を使用していない場合は、[ツール]、[インターネットオプション]、[詳細設定]、[リセット]の順に選択し、表示される指示に従います。
  - Firefox ESR (Extended Support Release)。
  - Chrome。

以下のオペレーティングシステムがサポートされています。

- Windows 10、Enterpriseエディション。これは、Standard VDAのみを使用するXenAppおよびXenDesktop 7.6 FP3 VDA

Standaloneインストールに適用されます。次のものはWindows 10上の7.6 FP3でサポートされません：

- HDX 3D Pro
- GPUアクセラレーション
- デスクトップコンポジションリダイレクト
- 従来のグラフィックモード
- セキュア起動
- VM Hosted Appsを使ったユニバーサルなWindowsアプリの公開
- Citrix Profile Management
- Windows 8.1、Professional、およびEnterpriseエディション。
- Windows 8、Professional、およびEnterpriseエディション。
- Windows 7 SP1、Professional、Enterprise、およびUltimateエディション。

サポートされているサーバーオペレーティングシステムでは、コマンドラインインターフェイスを使用してVDA for Windows Desktop OSをインストールしてサーバーVDI機能を使用できます。手順については、「[サーバーVDI](#)」を参照してください。

- Windows Server 2012 R2、Standard、およびDatacenterエディション。
- Windows Server 2012、Standard、およびDatacenterエディション。
- Windows Server 2008 R2 SP 1、Standard、Enterprise、およびDatacenterエディション。

要件：

- Microsoft .NET Framework 4.6.1。
- Microsoft .NET Framework 4.5.1 (4.5.2および4.6もサポートされます)。
- Microsoft .NET Framework 3.5.1 (Windows 7のみ)
- Microsoft Visual C++ 2005、2008、2010 Runtime (32ビットおよび64ビット)。これは、XenAppおよびXenDesktop 7.6、7.6 FP1およびFP2に適用されます。
- Microsoft Visual C++ 2008、2010、および2013のRuntime (32ビットおよび64ビット)。このアートの内容は、XenAppおよびXenDesktop 7.6 FP3 VDA Standaloneインストールに適用されます。

リモートPCアクセスでは、このVDAを社内の物理PC上にインストールします。

いくつかのマルチメディアアクセラレーション機能 (HDX MediaStream Windows Mediaリダイレクトなど) では、VDAのインストール先マシンにMicrosoft Media Foundationをインストールする必要があります。マシンにMedia Foundationがインストールされていない場合は、マルチメディアアクセラレーション機能がインストールされません。Citrixソフトウェアのインストール後にマシンからMedia Foundationを削除しないでください。これを削除すると、ユーザーがマシンにログオンでなくなります。Windows 8.1、Windows 8、およびWindows 7のほとんどのエディションにはMedia Foundationがインストールされており、削除することはできません。ただし、Nエディションには一部のメディア関連機能が付属しません。これらのソフトウェアは、Microsoft社またはサードパーティから入手できます。

VDAをインストールするときに、VDA for Windows Desktop OSのHDX 3D Proバージョンのインストールを選択できます。このバージョンは、DirectXやOpenGL指向のアプリケーション、およびビデオなどのリッチメディアに特に適しています。

以下のオペレーティングシステムがサポートされています。

- Windows Server 2012 R2、Standard、およびDatacenterエディション。
- Windows Server 2012、Standard、およびDatacenterエディション。
- Windows Server 2008 R2 SP 1、Standard、Enterprise、およびDatacenterエディション。

以下の必須ソフトウェアは、インストーラーにより自動的にインストールされます。これらのソフトウェアは、CitrixインストールメディアのSupportフォルダーにも収録されています。

- Microsoft .NET Framework 4.6.1。

- Microsoft .NET Framework 4.5.1 (4.5.2および4.6もサポートされます)。
- Microsoft .NET Framework 3.5.1 (Windows Server 2008 R2のみ)
- Microsoft Visual C++ 2005、2008、2010 Runtime (32ビットおよび64ビット)。これは、XenAppおよびXenDesktop 7.6、7.6 FP1およびFP2に適用されます。
- Microsoft Visual C++ 2008、2010、および2013のRuntime (32ビットおよび64ビット)。このアートの内容は、XenAppおよびXenDesktop 7.6 FP3 VDA Standaloneインストールに適用されます。

リモートデスクトップサービスの役割サービスが自動的にインストールされて有効になります。

いくつかのマルチメディアアクセラレーション機能 (HDX MediaStream Windows Mediaリダイレクトなど) では、VDAのインストール先マシンにMicrosoft Media Foundationをインストールする必要があります。マシンにMedia Foundationがインストールされていない場合は、マルチメディアアクセラレーション機能がインストールされません。Citrixソフトウェアのインストール後にマシンからMedia Foundationを削除しないでください。これを削除すると、ユーザーがマシンにログオンでなくなります。Windows Server 2012 R2、Windows Server 2012、およびWindows Server 2008 R2のほとんどのエディションでは、サーバーマネージャーによりMedia Foundation機能がインストールされます (Windows Server 2012 R2およびWindows Server 2012の場合はServerMediaFoundation、Windows Server 2008 R2の場合はDesktopExperience)。ただし、Nエディションには一部のメディア関連機能が付属しません。これらのソフトウェアは、Microsoft社またはサードパーティから入手できます。

サポートされるプラットフォームは、以下のとおりです。

- XenServer。
  - XenServer 6.5 SP1
  - XenServer 6.5
  - XenServer 6.2 SP1にHotfixを適用したもの (Hotfixを適用できるようにするには、SP1を適用しておく必要があります)
  - XenServer 6.1
- VMware vSphere - vSphere vCenterのリンクモードはサポートされません。
  - VMware vSphere 6.0 Update 1
  - VMware vSphere 6.0
  - VMware vSphere 5.5 Update 3
  - VMware vSphere 5.5 Update 2
  - VMware vSphere 5.5 Update 1
  - VMware vSphere 5.5
  - VMware vSphere 5.1 Update 3
  - VMware vSphere 5.1 Update 2
  - VMware vSphere 5.0 Update 3
  - VMware vSphere 5.0 Update 2
  - VMware vCenter 5.5 / 6アプライアンス
- System Center Virtual Machine Manager - これらのSystem Center Virtual Machine ManagerをサポートするHyper-Vのバージョンを含みます。
  - System Center Virtual Machine Manager 2012 R2
  - System Center Virtual Machine Manager 2012 SP1
  - System Center Virtual Machine Manager 2012
- Nutanix Acropolis 4.5 - このプラットフォームを使用している場合は、複数のXenAppおよびXenDesktop機能を実行できません。詳しくは、[CTX202032](#)を参照してください。Acropolisがある製品の使用方法については、<https://portal.nutanix.com/#/page/docs>を参照してください。

以下のクラウド環境がサポートされています。



- Amazon Web Services (AWS)
  - サポートされるWindowsサーバーOSで、アプリケーションやデスクトップをプロビジョニングできます。
  - AWSでは、SQL Server 2012 Enterprise Editionを使用できません。
  - AWSでは、デスクトップOSのインスタンスは提供されていません。
  - Amazon Relational Database Service (RDS) はサポートされません。
  - 詳しくは、AWSのドキュメントおよび「[Deploy XenApp and XenDesktop 7.5 and 7.6 with Amazon VPC](#)」を参照してください。
- Citrix CloudPlatform
  - Hotfix 4.2.1~4を適用したVersion 4.2.1以降がサポートされます。
  - ハイパーバイザーとして、XenServer 6.2 (Service Pack 1とXS62ESP1003適用済み) およびvSphere 5.1での動作がテストされています。
  - CloudPlatformでは、Hyper-Vハイパーバイザーはサポートされません。
  - CloudPlatform 4.3.0.1では、VMware vSphere 5.5がサポートされます。
  - そのほかのサポート情報およびLinuxシステムの要件については、CloudPlatformのドキュメント (使用するバージョンのCloudPlatformのリリースノートを含む) および「[XenApp and XenDesktop concepts and deployment on CloudPlatform](#)」を参照してください。

サポートされるハイパーバイザーについての最新情報については、[CTX131239](#)を参照してください。

Machine Creation Services、および仮想マシンへのActive Directoryアカウント設定機能は、以下の仮想化リソースおよびストレージ技術の組み合わせでサポートされます。推奨される組み合わせはアスタリスク (\*) で示します。

| 仮想化リソース   | ローカルディスク                       | NFS     | ブロックストレージ               | StorageLink |
|-----------|--------------------------------|---------|-------------------------|-------------|
| XenServer | はい                             | はい<br>* | はい                      | いいえ         |
| VMware    | はい (vMotionまたは動的な移行はサポートされません) | はい<br>* | はい                      | いいえ         |
| Hyper-V   | はい                             | いいえ     | はい* (クラスター共有ボリュームが必要です) | いいえ         |

リモートPCアクセスのWake on LAN機能を使用するには、Microsoft System Center Configuration Manager 2012が必要です。詳しくは、「[Configuration ManagerとリモートPCアクセスのWake on LAN](#)」を参照してください。

Active Directoryフォレストとドメインの以下の機能レベルがサポートされています。

- Windows 2000ネイティブ (ドメインコントローラーでサポートされない)
- Windows Server 2003
- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2012 R2

Windowsベースのユーザーデバイスまたはシンクライアントでの要件は以下のとおりです。

- DirectX 9

- Pixel Shader 2.0 (ハードウェアでのサポート)
  - 32ビット/ピクセル
  - 1.5GHz、32ビットまたは64ビットプロセッサ
  - 1GBのRAM
  - IGP (Integrated Graphics Processor) またはグラフィックカード上の128MBのビデオメモリ。
- HDXにより、必要なGPU機能が使用可能かどうかWindowsデバイスに照会され、使用できない場合は自動的にサーバー側のデスクトップコンポジションが使用されます。GPU機能が使用可能であってもプロセッサ速度やRAMの要件が満たされないデバイスは、デスクトップコンポジションリダイレクトの除外デバイスとしてGPOグループに追加してください。

最小使用可能帯域幅は1.5Mbpsです。推奨帯域幅は5Mbpsです。これらの値では、エンドツーエンドの遅延が考慮されていません。

Windows Mediaのクライアント側でのコンテンツ取得、Windows Mediaリダイレクト、およびリアルタイムWindows Mediaマルチメディアトランスコードでは、Receiver for Windows、Receiver for iOS、Receiver for Linuxの各クライアントがサポートされています。

Windows MediaコンテンツをWindows 8デバイス側で取得するには、デフォルトプログラムとしてCitrix Multimedia Redirectorを設定します。これを行うには、[コントロールパネル]、[プログラム]、[既定のプログラム]、[既定のプログラムの設定]の順に選択し、[Citrix Multimedia Redirector]を選択して[すべての項目に対し、既定のプログラムとして設定する]または[既定でこのプログラムで開く項目を選択する]のいずれかをクリックします。

GPUトランスコードでは、NVIDIA CUDAが有効なGPU (Compute Capability 1.1以上) が必要です。詳しくは、<http://developer.nvidia.com/cuda/cuda-gpus>を参照してください。

以下のクライアントとAdobe Flash Playerがサポートされています。

- Receiver for Windows (第2世代のFlashリダイレクト機能用) - 第2世代Flashリダイレクト機能ではAdobe Flash Player for Other Browser (「NPAPI (Netscape Plugin Application Programming Interface) Flash Player」と呼ばれることもあります)。
- Receiver for Linux (第2世代のFlashリダイレクト機能用) - 第2世代Flashリダイレクト機能ではAdobe Flash Player for other LinuxまたはAdobe Flash Player for Ubuntu。
- Citrix Online Plug-in 12.1 (従来のFlashリダイレクト機能) - 従来のFlashリダイレクト機能ではAdobe Flash Player for Windows Internet Explorer (「ActiveXプレーヤー」と呼ばれることもあります)。

ユーザーデバイス上のFlash Playerのバージョン (メジャーバージョン番号) は、サーバー上のものと同じまたはそれ以降である必要があります。ユーザーデバイスに以前のバージョンのFlash Playerがインストールされている、またはユーザーデバイスにFlash Playerがインストールされていない場合、Flashコンテンツはサーバー上で処理されます。

VDAを実行するマシンでの要件は以下のとおりです。

- Adobe Flash Player for Windows Internet Explorer (ActiveXプレーヤー)
- Microsoft Internet Explorer Version 7、8、9、10、11 (非Modern UIモード) Flashリダイレクトでは、サーバーにInternet Explorerが必要です。ほかのWebブラウザーを使用する場合、Flashコンテンツはサーバー側で処理されます。
- Internet Explorerの保護モードの無効化： ([ツール] > [インターネットオプション] > [セキュリティ] タブ > [保護モードを有効にする] チェックボックスをオフ)。変更を反映させるため、Internet Explorerを再起動します。

VDA for Windows Desktop OSをインストールするときに、HDX 3D Proバージョンのインストールを選択できます。

アプリケーションをホストする物理マシンまたは仮想マシンでは、GPUパススルーまたは仮想GPU (vGPU) を使用できます。

- GPUパススルー機能は、Citrix XenServerで使用できます。また、VMware vSphereおよびVMware ESXでも、同様の仮想 Direct Graphics Acceleration (vDGA) 機能を使用できます。
- vGPU機能は、Citrix XenServerで使用できます。[www.citrix.com/go/vGPU](http://www.citrix.com/go/vGPU)を参照してください (CitrixのMy Account資格情報が必要です)。

ホストコンピューターとして、4GB以上のRAMと2.3GHz以上の4つの仮想CPUを推奨します。

GPU (Graphical Processing Unit) :

- 無損失圧縮を含むCPUベース圧縮では、ホストコンピューター上のあらゆるディスプレイアダプター (配信するアプリケーションと互換性があるもの) がサポートされます。
- NVIDIA GRID APIを使用したGPUフレームバッファアクセスを最適化するには、HDX 3D ProでNVIDIA Quadroカードおよび最新のNVIDIAドライバーが必要です。NVIDIA GRIDでは高いフレームレートが配信されるため、ユーザーエクスペリエンスが向上します。
- HDX 3D ProでXenServerのvGPU機能を使用するための要件には、NVIDIA GRID K1およびK2カードが含まれています。

ユーザーデバイス :

- HDX 3D Proでは、ホストコンピューター上のGPUでサポートされるすべてのモニターの解像度がサポートされます。ただし、推奨されている最低限のユーザーデバイスおよびGPU仕様でのパフォーマンスを最適化するには、最大モニター解像度を1920×1200ピクセル (LAN接続の場合) または1280×1024ピクセル (WAN接続の場合) にすることをお勧めします。
- ユーザーデバイスでは、1GB以上のRAMと1.6GHz以上のCPUを推奨します。低帯域幅接続で必要とされるデフォルトの深圧縮コーデックを使用する場合は、より強力なCPUが必要です (ハードウェアでデコードしない場合)。パフォーマンスを最適化するには、ユーザーデバイスに2GB以上のRAMおよび3GHz以上のデュアルコアCPUを推奨します。
- マルチモニター環境の場合は、クアッドコアCPUを推奨します。
- HDX 3D Proで配信されたデスクトップやアプリケーションにアクセスする場合、ユーザーのデバイスに専用のGPUは必要ありません。
- Citrix Receiverのインストールが必要です。

サポートされるクライアント : Citrix Receiver for Windows、Receiver for Mac、Receiver for Linux

サポートされるビデオ会議アプリケーションは以下のとおりです。

- Citrix GoToMeeting HDFaces
- Adobe Connect
- Cisco WebEx
- IBM Sametime
- Microsoft Lync 2013および2010
- Microsoft Office Communicator
- Google+ハンガアウト
- Windows 8.x、Windows Server 2012、およびWindows Server 2012 R2上で動作するMedia Foundation形式のビデオアプリケーション
- Skype 6.7。WindowsクライアントでSkypeを使用するには、クライアント側およびサーバー側のレジストリを編集する必要があります。
  - クライアントのレジストリキー : HKEY\_CURRENT\_USER\Software\Citrix\HdxRealTime  
値の名前 : DefaultHeight、種類 : REG\_DWORD、値のデータ : 240  
  
値の名前 : DefaultWidth、種類 : REG\_DWORD、値のデータ : 320
  - サーバーのレジストリキー : HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\Vd3d\Compatibility

値の名前 : skype.exe、種類 : REG\_DWORD、値のデータ : 0

その他のユーザーデバイス要件 :

- サウンド再生のためのハードウェア
- DirectShow対応のWebカメラ (Webカメラのデフォルト設定を使用してください)。Webカメラ側のハードウェアエンコーディング機能を使用すると、クライアント側のCPU使用率が軽減されます。
- Webカメラの製造元から入手したドライバー (入手できる場合)

Receiver for WindowsおよびReceiver for Linux 13では、マルチストリームICAでのUDPオーディオがサポートされています。

Citrix Receiver for Windowsでは、エコーキャンセルがサポートされています。

- Universal Print Server - Universal Print Serverは、クライアント側およびサーバー側のコンポーネントで構成されています。UPClientコンポーネントは、VDAと一緒にインストールされます。UPServerコンポーネントはCitrixユニバーサルプリンタードライバーを使用する共有プリンターの各プリントサーバー上にインストールし、以下のオペレーティングシステムをサポートします。
  - Windows Server 2008 R2 SP1
  - Windows Server 2008 32ビット
  - Windows Server 2012 R2および2012。これは、XenAppおよびXenDesktop 7.6 FP3以降に適用されます。
- XenAppおよびXenDesktop 7.6 FP3。プリントサーバーにUPServerコンポーネントをインストールするには、次の前提条件が必要です。
  - Microsoft Visual Studio 2013 Runtime (32ビットおよび64ビット)
  - Microsoft .NET Framework 4.5.1。
  - 64ビットプラットフォーム用CDF\_x64.msiまたは32ビットプラットフォーム用CDF\_x86.msi
  - 64ビットプラットフォーム用UpsServer\_x64.msiまたは32ビットプラットフォーム用UpsServer\_x86.msi

セッションネットワークプリンターをプロビジョニングするXenAppおよびXenDesktopホスト上にインストールするUPClientコンポーネントは、Windows 10デスクトップと互換性があり、VDAインストールの一部です。

印刷操作の間にユーザー認証を実行するには、Universal Print ServerがリモートデスクトップサービスVDAと同じドメインに参加する必要があります。

- すべてのコンポーネントをこの製品リリースのインストールメディアに収録されているバージョンにアップグレードすることをお勧めします。
  - StoreFrontには2GBのメモリが必要です。システム要件については、StoreFrontのドキュメントを参照してください。この製品では、StoreFront 2.6以降がサポートされます。
  - Provisioning Servicesを併用する場合、Provisioning Services 7.0以降がサポートされます。
  - Citrixライセンスサーバーには40MBのディスク領域が必要です。システム要件については、ライセンスサーバーのドキュメントを参照してください。Citrix License Server for Windowsのみサポートされます。必要な最小バージョンは11.12.1です。
- Citrixポリシー情報をサイト構成データベースではなくActive Directoryに格納する場合、Microsoftグループポリシー管理インソール (GPMC) が必要です。詳しくは、Microsoft社のドキュメントを参照してください。
- VDAをインストールすると、デフォルトでReceiver for Windowsもインストールされます。その他のプラットフォームのシステム要件情報については、Receiver for Windowsのドキュメントを参照してください。
- Receiver for LinuxおよびReceiver for Macは、この製品のインストールメディアに収録されています。システム要件につ

ては、各製品のドキュメントを参照してください。

- Version 10.0よりも前のバージョンのAccess GatewayをこのリリースのXenDesktop/XenApp環境で使用する場合は、Windows 8.1およびWindows 8のクライアントはサポートされません。
- Desktop Lock - サポートされるオペレーティングシステム：
  - Windows 7 (Embedded Editionを含む)
  - Windows XP Embedded
  - Windows Vista

ユーザーデバイスがローカルエリアネットワーク (LAN) に接続されている必要があります。

サポートされるReceiver : Citrix Receiver for Windows Enterprise 3.4パッケージ以降。

- クライアントフォルダーのリダイレクト - サポートされるオペレーティングシステム：
  - サーバー : Windows Server 2008 R2 SP1、Windows Server 2012、Windows Server 2012 R2
  - クライアント (最新のCitrix Receiver for Windowsを使用する場合) : Windows 7、Windows 8、Windows 8.1
- 複数のネットワークインターフェイスカードがサポートされます。
- サポートされているバージョンについては、「[App-V](#)」を参照してください。

# 製品の技術概要

Jan 04, 2017

XenAppおよびXenDesktopによる仮想化ソリューションにより、IT担当者は仮想マシン、アプリケーション、ライセンス、およびセキュリティを完全に制御でき、あらゆるデバイスからのアクセスを提供できます。

XenAppおよびXenDesktopでは、以下の機能が提供されます。

- エンドユーザーは、デバイスで動作するオペレーティングシステムやインターフェイスに依存せずにアプリケーションやデスクトップを実行できます。
- 管理者はネットワークを管理して、特定のデバイスまたはすべてのデバイスにアクセスを提供したり制限したりできます。
- 管理者は、単一のデータセンターからネットワーク全体を管理できます。

XenAppとXenDesktopでは「FlexCast Management Architecture (FMA)」と呼ばれる共通の統合アーキテクチャが使用されます。FMAにより、単一サイトで複数のバージョンのXenAppまたはXenDesktopを実行でき、プロビジョニング機能が統合されます。

一般的なXenApp/XenDesktop環境は、いくつかの主要コンポーネントで構成されます。これらのコンポーネントは、ユーザーがアプリケーションやデスクトップにアクセスすると連係して動作し、サイトでのアクティビティデータをログとして記録します。

## Citrix Receiver

ユーザーデバイス上にインストールされるクライアントソフトウェアで、TCPポート80または443経由で仮想マシンに接続し、StoreFront Service APIを使用してStoreFrontと通信します。

## StoreFront

ユーザーを認証するインターフェイスであり、アプリケーションやデスクトップを管理してアプリケーションストアをホスします。StoreFrontは、XMLを使用してDelivery Controllerと通信します。

## Delivery Controller

XenAppおよびXenDesktopサイトの中心的管理コンポーネントで、リソース、アプリケーション、およびデスクトップを管理したりユーザー接続の負荷を最適化および分散したりするためのいくつかのサービスで構成されています。

## Virtual Delivery Agent (VDA)

Windowsのサーバーまたはデスクトップオペレーティングシステム (OS) のマシン上にインストールされるエージェントで、マシンやそのマシンでホストされるリソースをユーザーに提供します。WindowsサーバーOS上にインストールされたVDAは、複数ユーザーに対して複数の接続を提供します。これらの接続では、以下のいずれかのポートが使用されます。

- TCPポート80 (またはSSLが有効な場合は443)
- セッション画面の保持機能のためのCitrix Gateway Protocol (CGP) が有効な場合はTCPポート2598
- CGPが無効な場合、または従来のクライアントソフトウェアから接続する場合はTCPポート1494

## Broker Service

Delivery Controllerのサービスの1つで、ログオンしているユーザー、ログオン先、ユーザーのセッションリソース、既存のアプリケーションへの再接続が必要かどうかを追跡・監視します。Broker Serviceは、PowerShellコマンドを実行し、TCPポート80でBroker Agentと通信します。TCPポート443を使用するオプションはありません。

## Broker Agent

いくつかのプラグインをホストするエージェントで、リアルタイムデータを収集します。Broker AgentはVDA上で動作して、TCPポート80でControllerに接続します。TCPポート443を使用するオプションはありません。

## Monitor Service

Delivery Controllerのサービスの1つで、履歴データを収集してサイトのデータベース上に格納します。Monitor ServiceはTCPポート80または443で通信します。

#### ICAファイル/スタック

VDAへの接続に必要なユーザー情報が格納されています。

#### サイトデータベース

サイトのポリシー、マシンカタログ、およびデリバリーグループなど、Delivery Controllerのデータを格納するMicrosoft SQLデータベースです。

#### NetScaler Gateway

追加の資格情報によりLANのファイアウォール内外からのアクセスを保護するデータアクセスソリューションです。

#### Director

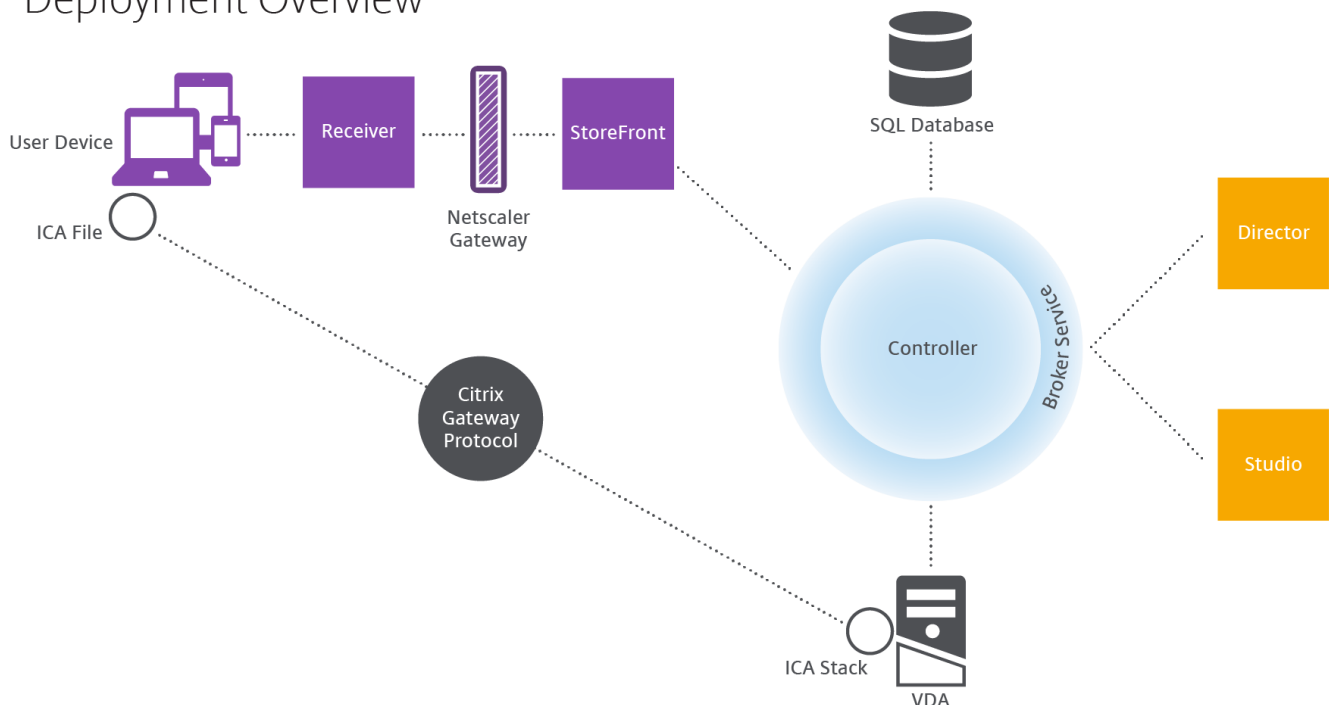
管理者がBroker Agentからのリアルタイムデータ、サイトデータベースからの履歴データ、およびNetScalerからのHDXデータにアクセスしてトラブルシューティングやサポート業務に使用するWebベースのツールです。Directorは、TCPポート80または443でControllerと通信します。

#### Studio

管理者がサイトの構成および管理を行うための管理コンソールで、Broker Agentからのリアルタイムデータにアクセスすることもできます。Studioは、TCPポート80でControllerと通信します。

XenAppおよびXenDesktopのサイトは、スケーラビリティ、高可用性、およびフェールオーバーを実現する特定の役割を持ついくつかのマシンで構成され、「セキュア・バイ・デザイン」なソリューションを提供します。XenApp/XenDesktopサイトは、VDAがインストールされたWindowsサーバーマシンとWindowsデスクトップマシン、およびアクセスを管理するDelivery Controllerで構成されます。

## Deployment Overview



VDAは、ユーザーがデスクトップやアプリケーションにアクセスすることを可能にするエージェントソフトウェアです。多くの場合、このコンポーネントはデータセンター内のサーバーまたはデスクトップマシン上にインストールされますが、リモートPCアクセス展開では社内の物理PC上にインストールされます。

Controllerは、リソース、アプリケーション、およびデスクトップを管理したりユーザー接続を最適化および負荷分散したりする、独立したいくつかのWindowsサービスで構成されます。各サイトでは1つまたは複数のControllerが動作し、理想的にはそのすべてが同一LAN上に配置されている必要があります。これは、セッションが遅延、帯域幅、およびネットワークの信頼性に依存するためです。

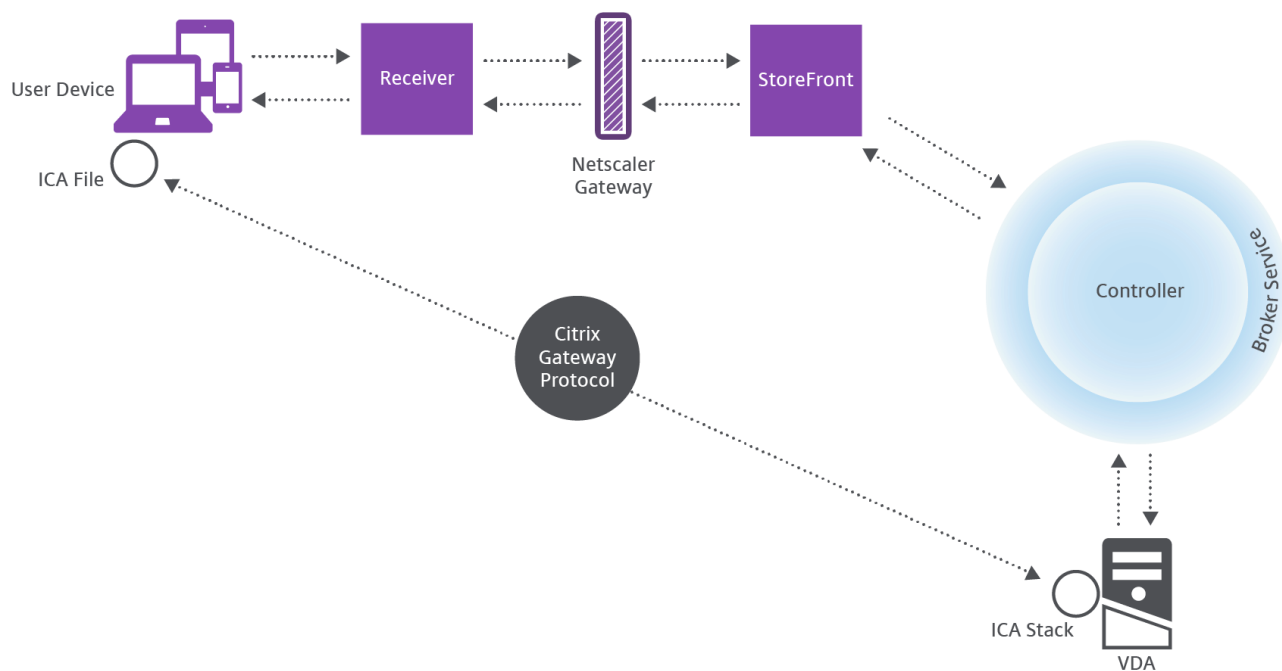
ユーザーがControllerに直接アクセスすることはありません。ユーザーとController間の通信の中継点としてVDAが機能します。ユーザーがStoreFrontを使用してサイトにログオンすると、ユーザーの資格情報がBroker Serviceに渡され、そこでそのユーザーに適用されるポリシーに基づいてプロファイルおよびリソースが取得されます。

ユーザーがXenAppまたはXenDesktopのセッションを開始するには、ユーザーデバイス上にインストールされたCitrix ReceiverまたはWebベースのReceiver for Web (RFW) を使用します。

ユーザーは、使用する物理デスクトップまたは仮想デスクトップ、または仮想アプリケーションをReceiverで選択します。

下図の経路で、Controllerにアクセスするためのユーザーの資格情報が転送されます。Controllerは、Broker Serviceと通信して必要なリソースを決定します。Receiverから送信される資格情報を暗号化で保護するために、StoreFront上にSSL証明書をインストールすることをお勧めします。

## User connections



Broker Serviceにより、ユーザーがアクセスできるデスクトップやアプリケーションが決定されます。

資格情報の検証後、アクセス可能なデスクトップやアプリケーションの情報がStoreFrontとReceiver経由でユーザー側に返送されます。ユーザーが一覧からアプリケーションまたはデスクトップを選択すると、その情報が同じ経路でControllerに送信され、そこでアプリケーションまたはデスクトップのホストとして適切なVDAが決定されます。

Controllerはユーザーの資格情報をメッセージとしてVDAに送信し、さらにユーザーと接続に関するすべてのデータをVDAに送信します。VDAが接続を受け入れて、同じ経路でReceiverに情報を返送します。Receiverがセッションで生成されたすべての情報を集めてIndependent Computing Architecture (ICA) ファイルを作成します。ICAファイルは、Receiverがローカルに



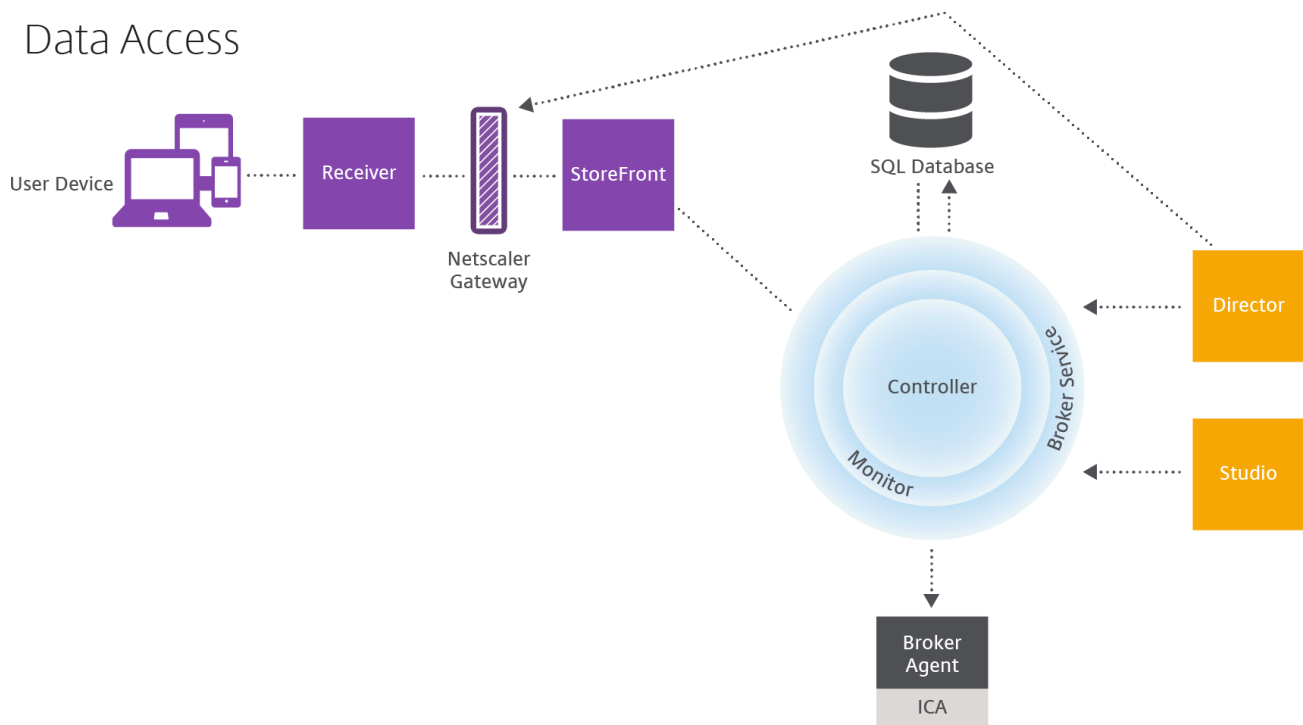
インストールされている場合はユーザーデバイス上で作成され、Webブラウザでアクセスしている場合はRFW上で作成されます。サイトが正しく構成されている場合、ユーザーの資格情報はこれらの処理をとおして暗号化されたまま転送されます。

ICAファイルがユーザーデバイスにコピーされ、VDA上で実行されるICAスタックとの直接接続が確立されます。この接続により、管理インフラストラクチャ (Receiver、StoreFront、およびController) がバイパスされます。

ReceiverとVDA間の接続ではCitrix Gateway Protocol (CGP) が使用されます。接続が中断されても、セッション画面の保持機能により同じVDAに再接続されます。管理インフラストラクチャ経由でセッションを再起動する必要はありません。セッション画面の保持機能の有効/無効の設定はStudioで行います。

クライアントがVDAに接続すると、ユーザーがログオンしたことがVDAによりControllerに通知されます。Controllerは、この情報をサイトデータベースに送信し、監視データベース内でのデータ記録を開始します。

IT担当者は、XenAppまたはXenDesktopの各セッションにより提供されるデータにStudioやDirectorでアクセスできます。Studioでは、Broker Agentからのリアルタイムデータにアクセスしてサイトを詳細に管理できます。Directorでは、Broker Agentからのリアルタイムデータに加えて、監視データベースに格納された履歴データやNetScaler GatewayからのHDXデータにアクセスしてヘルプデスク業務やトラブルシューティングに役立てることができます。



Controller内では、Broker Serviceがリアルタイムデータを提供する仮想マシン上の各セッションについてのセッションデータをレポートします。Monitor Serviceもこのリアルタイムデータを監視して、履歴データとして監視データベース内に格納します。

StudioはBroker Serviceとのみ通信するため、リアルタイムデータにのみアクセスします。Directorは、Broker Serviceと (Broker Agent内のプラグイン経由で) 通信してサイトデータベースにアクセスします。

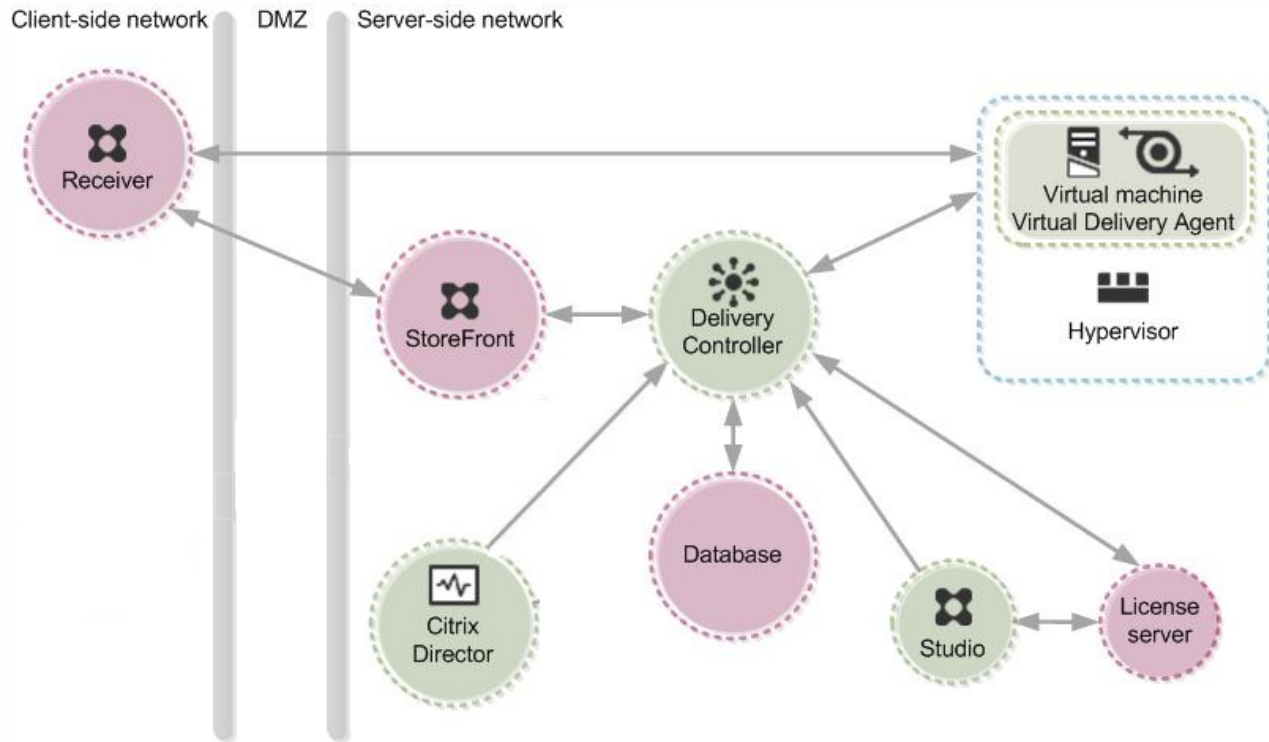
また、DirectorはNetScaler Gatewayにもアクセスして、HDXデータの情報を取得します。

- 概念およびコンポーネント
- Active Directory
- フォールトトレランス
- 配信方法

# 概念およびコンポーネント

Jan 04, 2017

この図は、一般的なXenApp/XenDesktop展開環境（「サイト」と呼ばれます）を構成する主要コンポーネントを示しています。



以下のコンポーネントで構成されます。

- **Delivery Controller** — Delivery Controllerは、あらゆるXenAppまたはXenDesktopサイトでの中心的な管理コンポーネントです。各サイトには1つ以上のDelivery Controllerが必要で、データセンター内で動作する1つ以上のサーバー上にインストールします。（サイトの信頼性および可用性を向上させるには、複数のサーバー上にControllerをインストールします）Controllerはいくつかのサービスで構成され、これらのサービスがハイパーバイザーと通信してアプリケーションやデスクトップを配信したり、ユーザーアクセスを認証および管理したり、ユーザーと仮想デスクトップやアプリケーションとの接続を仲介したり、接続を最適化して負荷を分散させたりします。各サービスのデータはサイトデータベースに格納されます。

Controllerは、仮想デスクトップの状態を管理してユーザーからの要求や管理構成に基づいてそれらを起動および停止します。一部のエディションでは、仮想化または物理的なWindows環境でユーザーの個人用設定を管理するためのProfile managementをインストールできます。

- **データベース** — XenAppまたはXenDesktopの各サイトでは、すべての構成情報やセッション情報を格納するためのMicrosoft SQL Serverデータベースが少なくとも1つ必要です。このデータベースには、Controllerを構成する各サービスによって収集および管理されたデータが格納されます。データセンター内にデータベースをインストールして、Controllerと永続的に接続されるようにしてください。
- **Virtual Delivery Agent (VDA)** — サイトからユーザーに提供する各物理マシンおよび仮想マシン上にVDAをインストールします。これにより、これらのマシンがControllerに登録され、ユーザーがこれらのマシンおよびマシン上でホストされるリソースを使用できるようになります。VDAは、マシンとユーザーデバイス間の接続を確立して、そのユーザーやセッ

セッションに必要なCitrixライセンスを検証して、適切なポリシーを適用します。VDAは、VDAに含まれているBroker Agentを介してController上のBroker Serviceとセッションに関する情報を送受信します。

XenAppおよびXenDesktopには、VDA for Windows Server OSおよびVDA for Windows Desktop OSが含まれています。VDA for Windows Server OSでは、同時に複数のユーザーがそのサーバーに接続できます。VDA for Windows Desktop OSでは、デスクトップへの単一ユーザー接続のみが許可されます。

- **StoreFront** — リソースをホストするサイトにアクセスするユーザーを認証して、ユーザーのデスクトップやアプリケーションのストアを管理します。StoreFrontにより、デスクトップやアプリケーションへのセルフサービスアクセスをユーザーに提供する「エンタープライズアプリケーションストア」がホストされます。また、ユーザーのアプリケーションサブスクリプション、ショートカット名、およびその他のデータが維持・管理され、さまざまなユーザーデバイスを使用するユーザーに一貫したアクセス体験が提供されます。
- **Receiver** — スマートフォン、タブレット、コンピューターなどのユーザーデバイスや仮想デスクトップなどのエンドポイント上にCitrix Receiverをインストールすると、アプリケーション、デスクトップ、および業務データへのすばやく安全なセルフサービスアクセスが提供されます。また、Windows、Web、およびSaaS (Software as a Service) アプリケーションへのオンデマンドアクセスも提供されます。Receiverをインストールできないデバイスでは、HTML5互換のWebブラウザからReceiver for HTML5を使用してアクセスすることもできます。
- **Studio** — 展開環境を構成して管理するための管理コンソールです。アプリケーション用とデスクトップ用に異なる管理コンソールを使用する必要はありません。Studioでは、環境のセットアップ、アプリケーションやデスクトップをホストするためのワークロードの作成、アプリケーションやデスクトップのユーザーへの割り当てなどを実行するさまざまなウィザードが提供されます。Studioでは、サイトのCitrixライセンスの割り当てや追跡も可能です。Studioには、Controller上のBroker Serviceからの情報が表示されます。

- **Director** — ITサポート担当者やヘルプデスクのスタッフは、このWebベースのツールを使ってシステムの状態を監視して、重大な障害が生じる前にトラブルシューティングを講じたりエンドユーザーをサポートしたりできます。Directorでは、複数のXenApp/XenDesktopサイトに接続して監視することができます。Directorには、以下のサービスからのセッション情報およびサイト情報が表示されます。

- Controller上のBroker Serviceからのリアルタイムセッションデータ。これには、VDA内のBroker AgentからBroker Serviceが収集したデータも含まれます。
- Controller上のMonitor Serviceからのサイト履歴データ。
- HDX Insightが収集したNetScalerからのHDXトラフィック（「ICAトラフィック」とも呼ばれます）に関するデータ。HDX Insightが付属するXenAppまたはXenDesktopのエディションおよびNetScalerを使用する環境に限られます。

また、Microsoft リモートアシスタンスを使用してユーザーのセッションを表示したり制御したりすることもできます。

- **ライセンスサーバー** — Citrix製品のライセンスの管理サーバーです。Controllerと通信して各ユーザーセッションのライセンスを管理し、Studioと通信してライセンスファイルを割り当てます。少なくとも1台のライセンスサーバーを作成して、ライセンスファイルを格納および管理する必要があります。
- **ハイパーバイザー** — サイトの仮想マシンをホストします。これには、アプリケーションやデスクトップをホストする仮想マシンだけでなく、XenAppやXenDesktopのコンポーネントをホストする仮想マシンも含まれます。ハイパーバイザーは、仮想マシンをホストする専用のコンピューター上にインストールします。XenAppおよびXenDesktopにはハイパーバイザーとしてCitrix XenServerが付属していますが、適切なバージョンのMicrosoft Hyper-VまたはVMware vSphereを使用することもできます。

多くのXenApp/XenDesktop展開ではハイパーバイザーを併用しますが、リモートPCアクセスを提供する場合、または仮想マシンのプロビジョニングにMCSではなくProvisioning Services (XenAppおよびXenDesktopの一部のエディションに付属)を使用する場合はハイパーバイザーは不要です。

一般的なXenAppまたはXenDesktop展開では、上図に示されていない以下の追加コンポーネントを使用することもできます。

- **Provisioning Services** — XenAppおよびXenDesktopの一部のエディションに付属するオプションコンポーネントです。

仮想マシンをプロビジョニングするMCSの代替として使用できます。MCSがマスターイメージのコピーを作成するのに対し、Provisioning Servicesはマスターイメージをユーザーデバイスにストリーム配信します。Provisioning Servicesではハイパーバイザーが不要なため、物理マシンをホストすることができます。サイト内のProvisioning Servicesは、Controllerと通信してユーザーにリソースを提供します。

- **NetScaler Gateway** — ユーザーが組織のファイアウォールの外側から接続する場合、Citrix NetScaler Gateway (旧称「Access Gateway」) 技術を使用して接続をSSLで保護できます。NetScaler GatewayやNetScaler VPX仮想アプライアンスは非武装地帯 (DMZ) に配置するSSL VPNアプライアンスで、企業ファイアウォールを介した安全な単一アクセスポイントを提供します。
- **Citrix CloudBridge** — 支店など遠隔地のユーザーがWANを介して仮想デスクトップに接続する環境では、Citrix CloudBridge (旧称「Citrix Branch Repeater」または「WANScaler」) 技術によりWAN接続のパフォーマンスを最適化できます。この製品によりWAN接続におけるパフォーマンスが向上するため、WAN接続でもLAN接続のようなユーザーエクスペリエンスが提供されます。CloudBridgeでは、さまざまなユーザー操作に優先順位を割り当てることができます。たとえば、ネットワーク上で大きなファイルや印刷ジョブを送信する操作に高い優先度を割り当てて、遠隔地のユーザーがストレスなく作業できるようにします。HDX WAN Optimization with CloudBridgeによりトークン化された圧縮およびデータ重複排除が提供され、帯域幅消費が減少してパフォーマンスが向上します。詳しくは、Citrix CloudBridgeのドキュメントを参照してください。

XenAppおよびXenDesktopでは、ユーザーに提供するリソースを「マシンカタログ」でセットアップし、そのリソースへのアクセスを「デリバリーグループ」で特定のユーザーに付与します。

## マシンカタログ

マシンカタログとは、単一のエンティティとして管理される物理マシンまたは仮想マシンのグループを指します。これらのシンおよびそのアプリケーションや仮想デスクトップは、ユーザーに提供する「リソース」です。マシンカタログ内のすべてのマシンには、同じオペレーティングシステムおよびVDAがインストールされている必要があります。また、同じアプリケーションまたは仮想デスクトップがそれらのマシン上で使用可能になっている必要があります。通常、管理者はマスターイメージを作成して、それを基にカタログ内に同一構成の仮想マシンを作成します。

管理者は、マシンカタログを作成するときにマシンの種類とプロビジョニング方法を指定します。

### マシンの種類

- **Windows Server OSマシン** — Windowsサーバーオペレーティングシステムが動作する仮想マシンまたは物理マシンで、XenApp公開アプリケーション（「サーバーでホストされるアプリケーション」とも呼ばれます）とXenApp公開デスクトップ（「サーバーでホストされるデスクトップ」とも呼ばれます）を配信します。これらのマシンには同時に複数のユーザーが接続できます。
- **WindowsデスクトップOSマシン** — Windowsデスクトップオペレーティングシステムが動作する仮想マシンまたは物理マシンで、「VDIデスクトップ」（Windowデスクトップオペレーティングシステムのデスクトップ。ユーザーによるパーソナライズも許可できる）、「VM Hosted Apps」（デスクトップオペレーティングシステム上のアプリケーション）、および物理マシンでホストされるデスクトップを配信します。これらの各デスクトップにアクセスできるのは単一のユーザーのみです。
- **リモートPCアクセス** — ホワイトリストに含まれている、オフィスにあるユーザーのコンピューターで、そのコンピューター上のリソースへのアクセスを、Citrix Receiverが動作するあらゆるデバイスに提供します。リモートPCアクセスでは、ユーザーの物理PCへのアクセスをXenDesktop展開で管理できます。

## プロビジョニング方法

- **Machine Creation Services (MCS)** — マスターイメージから仮想サーバーや仮想デスクトップをオンデマンドで作成する一連のサービスで、ストレージ使用を最適化したり、ユーザーのログオン時に仮想マシンを提供したりできます。Machine Creation Servicesは、Citrix Studioに完全に統合され管理されます。
- **Provisioning Services** — 単一の共有ディスクイメージからリアルタイムでコンピューターをプロビジョニングしたり再プロ

ビジョニングしたりすることができます。Provisioning Servicesでは、複数のターゲットデバイスを単一のデバイスコレクションとして管理します。デスクトップおよびアプリケーションは、マスターターゲットデバイスから作成されたProvisioning ServicesのvDiskで配信されます。これにより、物理ハードウェアまたは仮想マシンの処理能力を活かすことができます。Provisioning Servicesの管理は、Provisioning Servicesコンソールで行います。

- 既存のイメージ — データセンター内の仮想マシンに移行済みのデスクトップやアプリケーションがこれにあたります。管理者は、サードパーティ製のESD (Electronic Software Delivery : 電子ソフトウェア配信) ツールを使用してターゲットデバイスを個別または集合的に管理します。

### デリバリーグループ

デリバリーグループとは、共通のリソースへのアクセスが付与されたユーザーのグループを指します。デリバリーグループは、マシンカタログのマシンと、サイトにアクセスできるActive Directoryユーザーが属しています。通常、デリバリーグループにユーザーを割り当てるにはActive Directoryグループを使用します。これは、Active Directoryのグループもデリバリーグループも同様の要件に基づいてユーザーをグループ化するためのものだからです。

1つのデリバリーグループに複数のマシンカタログのマシンを追加したり、1つのマシンカタログのマシンを複数のデリバリーグループに追加したりすることができます。ただし、特定のマシンを同時に複数のデリバリーグループに追加することはできません。管理者は、アプリケーション、デスクトップ、またはその両方を配信するデリバリーグループを作成できます。管理者は、デリバリーグループ内のユーザーがどのリソースにアクセスできるかを定義します。たとえば、異なるアプリケーションを異なるユーザーに配信する場合、マシンカタログのマスターイメージ上にそれらのアプリケーションをすべてインストールしておき、複数のデリバリーグループに追加するための十分な数のマシンをそのカタログに作成できます。次に、それらのマシンから適切なアプリケーションが適切なユーザーに配信されるように各デリバリーグループを構成します。

XenApp 6.5やそれ以前のバージョンのXenAppの運用経験がある場合は、これらのバージョンとXenApp/XenDesktop 7.6との違いについて理解しておくくと便利です。

以前のバージョンとXenApp/XenDesktop 7.6との機能や用語の対応は次の表のとおりです。ただし、これらは単なる名前の置き換えではなく、機能的にも更新されていることに注意してください。

| XenApp 6.5以前の機能                           | 対応するXenApp/XenDesktop 7.6の機能                                                     |
|-------------------------------------------|----------------------------------------------------------------------------------|
| Independent Management Architecture (IMA) | FlexCast Management Architecture (FMA)                                           |
| ファーム                                      | サイト                                                                              |
| ワーカーグループ                                  | マシンカタログ<br>Delivery Group                                                        |
| ワーカー                                      | Virtual Delivery Agent (VDA)<br>サーバーOSマシン、サーバーOS VDA<br>デスクトップOSマシン、デスクトップOS VDA |
| リモートデスクトップサービス (RDS) またはターミナルサービスマシン      | サーバーOSマシン、サーバーOS VDA                                                             |

| XenApp 6.5以前の機能 | 対応するXenApp/XenDesktop 7.6の機能  |
|-----------------|-------------------------------|
| デリバリーサービスコンソール  | Citrix StudioとCitrix Director |
| アプリケーションの公開     | アプリケーションの配信                   |
| データストア          | データベース                        |
| 負荷評価基準          | 負荷管理のポリシー                     |
| 管理者             | 委任管理者<br>役割<br>Scope          |

XenApp/XenDesktop 7.6は「FlexCast Management Architecture (FMA)」と呼ばれるアーキテクチャに基づいています。FMAは、Citrixテクノロジ間の相互運用性と管理モジュール性を可能にする、サービス指向のアーキテクチャです。FMAにより、アプリケーション配信、モビリティ、サービス、フレキシブルなプロビジョニング、およびクラウド管理のためのプラットフォームが提供されます。

FMAは、XenApp 6.5以前で使用されていたIMA (Independent Management Architecture) を置き換えるものです。

FMAの主要要素とXenApp 6.5以前のバージョンとの違いは以下のとおりです。

#### デリバリーサイト

XenApp 6.5以前のバージョンでは、最上位レベルのオブジェクトが「ファーム」と呼ばれていました。XenApp/XenDesktop 7.6での最上位レベルは「デリバリーサイト」です。デリバリーサイトでは、ユーザーのグループにアプリケーションやデスクトップが提供されます。

FMAでは、サイトを展開する管理者がドメインに属している必要があります。たとえばサーバーをインストールするには、アカウントにローカル管理者権限があり、Active Directoryのドメインユーザーである必要があります。

#### マシンカタログとデリバリーグループ

XenApp 6.5以前のバージョンでは、アプリケーションやサーバーソフトウェアを効率的に管理するため、ホストマシンがワーカーグループとしてグループ化されました。管理者は、ワーカーグループ内のすべてのマシン上のアプリケーションや負荷分散を単一エンティティとして管理できました。また、アプリケーションやマシンをフォルダーで分類することもできました。

XenApp/XenDesktop 7.6では、マシンカタログとデリバリーグループを使用して、マシン、負荷分散、およびアプリケーションやデスクトップを管理します。

#### Virtual Delivery Agent

XenApp 6.5以前のバージョンでは、ワーカーグループ内のワーカーマシンがユーザーのアプリケーションをホストして、データコレクターと通信しました。XenApp/XenDesktop 7.6では、VDAがDelivery Controllerと通信して、Delivery Controllerがユーザー接続を管理します。

#### Delivery Controller

XenApp 6.5以前のバージョンでは、ユーザーからの接続要求とハイパーバイザーとの通信をゾーンマスターが担当していました。XenApp/XenDesktop 7.6では、接続要求はサイト内の各Controllerに分散されて処理されます。

XenApp 6.5以前のバージョンでは、ゾーンによりWAN接続を介したサーバー集約およびデータ複製が提供されていました。XenApp/XenDesktop 7.6にはゾーンに相当するものはありませんが、WANを介した複数の場所からアプリケーションをユーザーに提供できます。管理者は、特定の地域やデータセンターのデリバリーサイトを作成して、ユーザーが複数のデリバリーサイトにアクセスできるように構成できます。XenApp/XenDesktop 7.6でApp Orchestrationを使用すると、地理的に異なる場所の複数のサイトを管理できます。

### Citrix StudioとCitrix Director

Studioコンソールでは、環境を構成したり、ユーザーにアプリケーションやデスクトップへのアクセスを提供したりできます。Studioは、XenApp 6.5以前のバージョンのデリバリーサービスコンソールに相当するものです。

また、管理者はDirectorを使用して環境の監視、ユーザーセッションのシャドウ、およびトラブルシューティングを行います。ユーザーのセッションをシャドウするには、Microsoftリモートアシスタンスが有効になっている必要があります (VDAのインストール時に自動的に有効になります)。

### アプリケーションの配信

XenApp 6.5以前のバージョンでは、アプリケーションの公開ウィザードを使用してアプリケーションをユーザーに公開しました。XenApp/XenDesktop 7.6では、Studioを使ってアプリケーションを作成および追加して、配信先のユーザーのデリバリーグループを選択します。Studioでは、最初にサイトを構成し、マシンカタログを作成および指定してから、そのマシンカタログにデリバリーグループを作成します。デリバリーグループでは、アプリケーションへのアクセスを許可するユーザーを指定します。

### データベース

XenApp/XenDesktop 7.6の構成情報は、IMAデータストアに格納されません。代わりに、構成情報およびセッション情報のデータストアとしてMicrosoft SQL Serverデータベースが使用されます。

### 負荷管理のポリシー

XenApp 6.5以前のバージョンの負荷評価基準では、事前定義された値に基づいてマシンの負荷が評価されていました。これにより、より負荷の軽いマシンにユーザー接続が割り当てられました。

XenApp/XenDesktop 7.6では、負荷管理ポリシーを使用してマシン間の負荷を管理します。

### 委任管理者

XenApp 6.5以前のバージョンでは、カスタム管理者を作成してフォルダーやオブジェクト単位で管理権限を割り当てることできました。XenApp/XenDesktop 7.6では、役割とスコープのペアを使用してカスタム管理者の権限を定義します。役割はその管理者が担当する管理タスクに対応しており、それらに割り当てられた権限を使用して管理業務を委任できます。スコープは、接続、マシン カタログ、デリバリーグループなど、その管理者が管理できるオブジェクトをグループ化したものです。組み込みの役割には、ヘルプデスク、アプリケーション、ホスト、カタログなど特定の権限が事前定義されています。たとえば、ヘルプデスク管理者は特定のサイト上の個々のユーザーのみを対象としますが、すべての管理権限を実行できる管理者は展開環境全体を監視して、システム全体の問題を解決できます。

FMAへの移行により、XenApp 6.5以前のバージョンで提供されていたいくつかの機能の実装が変更されたり、ほかの機能、コンポーネント、またはツールに置き換えられたりしています。

| XenApp 6.5以前の機能                | 対応するXenApp/XenDesktop 7.6の機能                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| セッションの事前起動とセッション残留はポリシーで構成します。 | <p>セッションの事前起動とセッション残留はデリバリーグループを編集して構成します。</p> <p>これらの機能では、セッションが要求される前にセッションを開始したり (セッションの事前起動)、ユーザーがすべてのアプリケーションを閉じた後もセッションをアクティブな状態で保持したり (セッション残留) できます。これにより、ユーザーがアプリケーションにすばやくアクセスできるようになります。XenApp 6.5以前のバージョンでは、ポリシーを使用してこれらの機能を構成していました。XenApp/XenDesktop 7.6では、既存のデリバリーグループを編集することでこれらの機能を有効にします。この設定は、そのデリバリーグループに含まれているユーザーに適用されます。「<a href="#">セッションの事前起動およびセッション残留の構成</a>」を参照してください。</p> |



|                                                                                                                             |                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>XenApp 6.5以<br/>前の機能<br/>認証が不要な<br/>ユーザー（匿<br/>名ユーザー）<br/>のサポートは<br/>公開アプリ<br/>ケーションの<br/>プロパティを<br/>設定して構成<br/>します。</p> | <p>対応するXenApp/XenDesktop 7.6の機能<br/>認証が不要なユーザー（匿名ユーザー）のサポートはデリバリーグループのユーザープロパティを設定して構成します。「<a href="#">ユーザー</a>」を参照してください。</p>                              |
| <p>ローカルホス<br/>トキャッシュ<br/>により、デー<br/>タストアに接<br/>続できない状<br/>態でもワー<br/>カーサーバー<br/>が正しく動作<br/>します。</p>                          | <p>接続リリースにより、サイトデータベースに接続できない状態でも、ユーザーが最近使用していたアプリケーションやデスクトップに接続および再接続できます。接続リリースは、SQL Server高可用性ベストプラクティスを補完する機能です。「<a href="#">接続リリース</a>」を参照してください。</p> |
| <p>アプリケー<br/>ションのスト<br/>リーム配信</p>                                                                                           | <p>App-Vによるアプリケーションのストリーム配信をStudioで管理できます。</p>                                                                                                              |
| <p>Web Interface</p>                                                                                                        | <p>StoreFrontへの移行をお勧めします。</p>                                                                                                                               |
| <p>SmartAuditor</p>                                                                                                         | <p>構成ログには、管理者視点のすべてのセッションアクティビティが記録されます。または、サードパーティ製のCitrix準拠のセッション記録ツールを使用できます。</p>                                                                        |

# Active Directory

Jan 04, 2017

認証および承認にはActive Directoryが使用されます。Active DirectoryのKerberosインフラストラクチャにより、Delivery Controllerとの通信の機密性および整合性が保護されます。Kerberosについて詳しくは、Microsoft社のドキュメントを参照してください。

「[システム要件](#)」で、フォレストとドメインでサポートされる機能レベルについて確認してください。ポリシーのモデル作成機能を使用するには、ドメインコントローラーでWindows Server 2003～Windows Server 2012 R2が動作している必要があります（ドメインの機能レベルには影響しません）。

以下の環境がサポートされています。

- ユーザーアカウントおよびコンピューターアカウントが単一Active Directoryフォレスト内のドメインに属している。同一フォレスト内であれば、ユーザーアカウントとコンピューターアカウントが異なるドメインに属していても構いません。このような環境では、すべてのドメイン機能レベルおよびフォレスト機能レベルがサポートされます。
- ユーザーアカウントが、Controllerおよび仮想デスクトップのコンピューターアカウントと異なるActive Directoryフォレストに属している。このような環境では、Controllerおよび仮想デスクトップのコンピューターアカウントのドメインが、ユーザーアカウントのドメインを信頼している必要があります。フォレストの信頼または外部の信頼を使用できます。このような環境では、すべてのドメイン機能レベルおよびフォレスト機能レベルがサポートされます。
- Controllerのコンピューターアカウントが、仮想デスクトップのコンピューターアカウントが属している追加のActive Directoryフォレストと異なるフォレストに属している。このような環境では、Controllerのコンピューターアカウントのドメインと、仮想デスクトップのコンピューターアカウントのすべてのドメインとの間に相互信頼関係が必要です。このような環境では、Controllerまたは仮想デスクトップのコンピューターアカウントが属しているすべてのドメインが [Windows 2000ネイティブ] 機能レベルまたはそれ以上である必要があります。すべてのフォレスト機能レベルがサポートされます。
- 書き込み可能なドメインコントローラー。読み取り専用ドメインコントローラーはサポートされません。

必要に応じて、Virtual Delivery Agent (VDA) で登録可能なControllerを検出するときに、Active Directoryの情報を使用することもできます。この機能は主に後方互換性を保持するためのもので、VDAとControllerが同じActive Directoryフォレストに属している場合のみ使用できます。この検出方法について詳しくは、「[Active Directory OUベースのController検出](#)」および[CTX119945](#)を参照してください。

注：このトピックの内容は、XenDesktop 7.1、およびXenApp 7.5以降に適用されます。これらの製品の以前のバージョンには適用されません。

複数のフォレストがあるActive Directory環境では、一方向または双方向の信頼関係が構成済みの場合にDNSフォワーダーによる名前参照や登録を使用できます。適切なActive Directoryユーザーがコンピューターアカウントを作成できるようにするには、オブジェクト制御の委任ウィザードを使用します。このウィザードについて詳しくは、Microsoft社のドキュメントを参照してください。

適切なDNSフォワーダーがフォレスト間に存在する場合、DNSインフラストラクチャにDNS逆引きゾーンは必要ありません。

VDAとControllerが別のフォレストにある場合、Active DirectoryとNetBIOSの名前が異なっているかどうかに関係なく、レジストリキーSupportMultipleForestが必要です。SupportMultipleForestキーは、VDA上でのみ必要です。以下のレジストリキーを追加してください。

注意：レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディ

ターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

- HKEY\_LOCAL\_MACHINE\Software\Citrix\VirtualDesktopAgent\SupportMultipleForest
  - 値の名前：SupportMultipleForest
  - 種類：REG\_DWORD
  - 値のデータ：0x00000001 (1)

DNS名前空間がActive Directoryのそれと異なる場合、DNS逆引き構成が必要になることがあります。

セットアップ時に外部信頼が構成済みの場合は、レジストリキーListOfSIDsが必要になります。また、Active DirectoryのFQDNがDNS FQDNと異なる場合、またはドメインコントローラーのドメインがActive Directory FQDNとは異なるNetBIOS名を持っている場合も、レジストリキーListOfSIDsが必要です。以下のレジストリキーを追加します。

- 32ビットまたは64ビットのVDA：HKEY\_LOCAL\_MACHINE\Software\Citrix\VirtualDesktopAgent\ListOfSIDs
  - 値の名前：ListOfSIDs
  - 種類：REG\_SZ
  - 値のデータ：Controllerのセキュリティ識別子 (SID)

適切な外部信頼が構成済みの場合、VDA上で以下の変更を行います。

1. \Citrix\Virtual Desktop Agent\brokeragentconfig.exe.configファイルを検索します。
2. ファイルのバックアップコピーを作成します。
3. メモ帳などのテキストエディターを使ってファイルを開きます。
4. 「allowNtlm="false"」を「allowNtlm="true"」に変更します。
5. ファイルを保存します。

ListOfSIDsレジストリキーを追加してbrokeragent.exe.configファイルを編集したら、Citrix Desktop Serviceを再起動して変更を適用します。

次の表は、サポートされる信頼の種類を示しています。

| 信頼の種類   | 推移性        | 方向        | このリリースでのサポート |
|---------|------------|-----------|--------------|
| 親および子   | 推移的        | 双方向       | はい           |
| ツリールート  | 推移的        | 双方向       | はい           |
| 外部      | 非推移的       | 一方向または双方向 | はい           |
| フォレスト   | 推移的        | 一方向または双方向 | はい           |
| ショートカット | 推移的        | 一方向または双方向 | はい           |
| 領域      | 推移的または非推移的 | 一方向または双方向 | いいえ          |

複雑なActive Directory環境での展開について詳しくは、[CTX134971](#)を参照してください。

# フォールトトレランス

Jan 04, 2017

このドキュメントでは、ビジネスクリティカルなアプリケーションやデスクトップにユーザーが常にアクセスできるように、サイトのフォールトトレランスを強化するための方法について概説します。

すべての情報は、サイト構成データベースに保存されます。Delivery ControllerはほかのControllerとは通信せず、サイト構成データベースとのみ通信します。任意のControllerをシャットダウンしても、そのサイトのほかのControllerには影響しません。しかしながら、これはサイト構成データベースが単一障害点になりうることを意味します。このデータベースサーバー障害が発生しても、仮想デスクトップに対する既存の接続はユーザーが仮想デスクトップからログオフまたは切断するまで機能し続けます。ただし、データベースサーバーが使用できるようになるまで、新しい接続は拒否されます。

データベースのバックアップを定期的に作成して、データベースサーバーに障害が発生してもバックアップから復元できるようにすることをお勧めします。さらに、自動フェールオーバーのための以下の高可用性ソリューションがあります。

- SQLミラーリング — 推奨されるソリューションです。データベースをミラーリングすると、アクティブなデータベースサーバーが停止しても自動的にフェールオーバーされるため、ユーザーへのサービスが中断されることはありません。ただし、各データベースサーバー上に完全なSQL Serverライセンスが必要になるため、ほかのソリューションよりも費用が高くなります。SQL Server Expressエディションを使用してデータベースをミラーリングすることはできません。
- ハイパーバイザーの高可用性機能の使用 — この方法では、データベースを仮想マシンとして展開して、ハイパーバイザー側の高可用性機能を使用します。このソリューションでは既存のハイパーバイザーソフトウェアを使用でき、またSQL Expressも使用できるため、ミラーリングよりも費用が安いというメリットがあります。ただし、データベースの新しい仮想マシンの起動に時間がかかるため、自動フェールオーバー処理が遅くなり、ユーザーへのサービスが中断する可能性があります。
- SQLクラスタリング — MicrosoftのSQLクラスタリングテクノロジーを使用して、任意のサーバーに障害が起きた場合に5のサーバーが自動的にタスクや実行内容を引き継ぐようにできます。ただし、このソリューションのセットアップは複雑で、SQLミラーリングなどほかのソリューションよりも自動フェールオーバー処理には一般的に時間がかかります。
- AlwaysOn可用性グループ機能は、SQL Server 2012で導入されたエンタープライズレベルの高可用性および障害回復ソリューションで、これによってユーザーデータベースの可用性が最大になります。AlwaysOn可用性グループ機能では、Windows Server Failover Clustering (WSFC) ノード上にSQL Serverインスタンスが存在する必要があります。詳しくは、「[AlwaysOn Availability Groups \(SQL Server\)](#)」を参照してください。

注：SQLクラスタ化またはSQLミラー化インストールにおけるノード上へのControllerのインストールはサポートされていません。

この構成手順には、サイトを作成する前にSQL Server管理ツールを使って管理者が実行するタスクが含まれます。残りのタスクは、管理者がサイトを作成するウィザードを実行するときに行います。

ミラー環境には2つ以上のSQL Serverマシンが必要です（ここではSQL Server AとSQL Server Bとします）。プライマリおよびセカンダリ（ミラー）でSQL Server Expressエディションを使用することはできません。

Microsoft SQL Server管理ツールを使って、SQL Serverデータベースを構成します。

1. SQL ServerソフトウェアをSQL Server AおよびSQL Server Bにインストールします。
2. SQL Server Aで、プライマリとして使用するデータベースを作成します（myDatabaseMirrorなど）。
  - データベースは簡易モデルではなく完全復旧モデルを使用する必要があります（デフォルトでは簡易モデルが構成されますが、これではトランザクションログがバックアップされません）。
  - データベースを作成するときは、照合順序設定としてLatin1\_General\_100\_CI\_AS\_KS（Latin1\_Generalは国により異なる）

ります。例：Japanese\_100\_CI\_AS\_KS) を使用します。データベースの作成時にこの照合順序設定を指定しないと、そのデータベースでのサービスのスキーマ作成に失敗し、「<service>：スキーマには大文字と小文字を区別しないデータベースが必要です」というエラーメッセージが表示されます (<service>は作成しているスキーマのサービス名)。

- [CTX139552](#)の説明に従って、Read-Committedスナップショットを有効にします。エラーを避けるには、データベースのミラーリングを開始する前にこのスナップショットを有効する必要があります。
3. SQL Server Aで、データベースをファイルにバックアップして、それをSQL Server Bにコピーします。
  4. SQL Server Bで、バックアップファイルをSQL Server Bに復元します。
  5. SQL Server Aでミラーリングを開始します。

次の手順は、Citrix管理者（サイトを作成するウィザードを実行する管理者）に完全なデータベース権限があるかどうかにより異なります。

- Citrix管理者にデータベース権限がある（データベース管理者とCitrix管理者が同じ人物である）場合、Studioによりすべての処理が実行されます。
  1. Citrix管理者はStudioを使ってサイトを作成し、作成済みのSQL Server Aデータベースのアドレスとその名前（myDatabaseMirror）を指定します。
  2. データベーススクリプトが自動的に適用され、プライマリおよびミラーのデータベースが設定されます。
- Citrix管理者にデータベース権限がない場合は、データベース管理者に支援を依頼する必要があります。
  1. Citrix管理者はStudioを使ってサイトを作成し、作成済みのSQL Serverのアドレスとその名前（myDatabaseMirror）を指定します。
  2. サイトを作成するウィザードで [スクリプトの生成] をクリックしてミラースクリプトおよびプライマリスクリプトを生成します。Citrix管理者はこれらのスクリプトをデータベース管理者に提供し、データベース管理者がスクリプトを実行します（最初にミラースクリプトを実行する必要があります）。データベース管理者はタスクが完了したらCitrix管理者にそれを通知する必要があります。
  3. Citrix管理者はStudioに戻り、サイトを作成するウィザードの残りの手順を完了させます。プライマリデータベースおよびミラーデータベースが設定されます。

サイトの作成後にミラーリング設定を検証するには、PowerShellコマンドレット `Get-ConfigDbConnection` を実行して、ミラーに対する接続文字列でフェールオーバーパートナーが設定されていることを確認します。

ミラー化されたデータベース環境でDelivery Controllerを後から追加、移動、または削除する場合、またはVDAを移動する場合は、検討事項について「[Controllerの追加、削除、または移動](#)」を参照してください。

サイト内のすべてのDelivery Controllerで障害が起きた場合、Virtual Delivery Agentが高可用性モードで動作するように構成できます。これにより、ユーザーは障害発生後もデスクトップやアプリケーションにアクセスして使用することができます。i高可用性モードでは、Controllerを経由しない、VDAへの直接ICA接続が可能になります。

この機能は、すべてのControllerとの通信に障害がある場合にのみ使用してください。ほかの高可用性ソリューションの代替策ではありません。詳しくは、[CTX127564](#)を参照してください。

SQL Server高可用性ベストプラクティスを補完する接続リース機能を使用すると、サイトデータベースが使用不可の場合でも、ユーザーが直近で使用したアプリケーションやデスクトップに接続および再接続できるようになります。詳しくは、「[接続リース](#)」を参照してください。

# 配信方法

Jan 04, 2017

単一の仮想化環境ですべてのユーザーのニーズを満たすことは容易ではありません。XenAppおよびXenDesktopの管理者は、さまざまな方法（「FlexCast」と呼ばれます）を使用してユーザーエクスペリエンスをカスタマイズできます。

これらの配信方法にはそれぞれ一長一短があり、ユーザーの使用方法に応じて最適なユーザーエクスペリエンスを提供できません。

今日、スマートフォンやタブレットなどのタッチスクリーンデバイスが移動環境として一般的に使用されています。これらのデバイス上で、大きな画面と従来のポインティングデバイスでの使用が前提となっているWindowsベースのアプリケーションを操作する場合、問題が生じることがあります。

XenAppとCitrix Receiverによるソリューションでは、Windowsベースのアプリケーションを修正することなく、すべての機能を安全にモバイルユーザーに提供できます。

XenAppの公開アプリケーションによる配信方法では、HDX MobileテクノロジーによりWindowsアプリケーションがモバイル環境に最適化されます。この方法では、Windowsアプリケーションがタッチスクリーン上で操作できるようになり、マルチタッチジェスチャ、ネイティブのメニューコントロール、カメラ、およびGPS機能がサポートされます。XenAppおよびXenDesktopでは、アプリケーションのソースコードを変更しなくても、多くのタッチ機能がそのまま使用可能になります。

以下のタッチ操作が含まれます。

- 編集フィールド操作時にキーボードが自動的に表示されます。
- Windowsコンボボックスコントロールの代わりにタッチ操作に適したピッカーコントロールが表示されます。
- ピンチやズームなどのマルチタッチジェスチャを使用できます。
- 慣性スクロールが適用されます。
- タッチパッドまたはカーソル操作でナビゲーションできます。

多くの業種で3年から5年ごとに必要になる物理マシンの更新は面倒な作業であり、最新のオペレーティングシステムやアプリケーションが必要な部署では特に大掛かりになりがちです。また、組織の成長による新しいマシンの導入に伴う諸経費も軽視できません。

VDI Personal vDiskを使用した配信方法では、ユーザーによる完全なパーソナル設定が可能なデスクトップオペレーティングシステムをサーバーリソースからシンクライアントを含むさまざまなマシンに提供できます。管理者は、プロセッサ、メモリ、およびストレージなどのリソースをネットワーク上のデータセンターで提供する仮想マシンを作成できます。

これにより、古いマシンの使用を延長しながら最新のソフトウェアを維持して、アップグレード時のダウンタイムを最小限に抑えることができます。

ネットワークセキュリティの問題はますます重要になっており、契約社員やパートナーなどの社外作業者に社内のアプリケーションやデータへのアクセスを許可する場合は特に慎重に行う必要があります。作業者にラップトップなどのデバイスを貸与する場合も、セキュリティに対する配慮が必要です。

XenDesktopおよびXenAppでは、データ、アプリケーション、およびデスクトップがファイアウォールで保護された安全なネットワーク内に格納されます。エンドユーザーが転送する情報は、画面の更新、マウスのクリック、キーボード入力のみ

す。これらのリソースをデータセンター内に保持することで、XenDesktopおよびXenAppは一般的なSSL VPNよりも安全なリモートアクセスソリューションを提供します。

Personal vDiskを使用したVDI展開では、ネットワークサーバー上に仮想マシンを作成し、シングルユーザーデスクトップオペレーティングシステムをシンクライアントやユーザーの個人所有デバイスに配信できます。これにより、高価な装置を購入しなくても、社外作業者のセキュリティの問題に対処できます。

新しいオペレーティングシステムに移行する場合、従来のアプリケーションやそのオペレーティングシステムをサポートしないアプリケーションの処置が課題になります。

アプリケーションを仮想マシンでホストすることで、ユーザーはアップグレード済みの仮想マシン上のCitrix Receiverを使用して、従来のアプリケーションを正しく実行できます。この間にIT部署は古いアプリケーションの問題を検証および解決でき、新しいオペレーティングシステムへのユーザーの移行を容易にして、ヘルプデスクへの問い合わせを効率化できます。

さらに、移行時にXenDesktopを使用することで、以下のメリットが生じます。

- デスクトップに関する複雑さが軽減する。
- IT部署が環境をより詳細に制御できるようになる。
- エンドユーザーの使用デバイスや作業場所についての柔軟性が向上する。

多くの設計事務所や製造会社では、高度な3Dグラフィックアプリケーションが使用されています。これらの組織では、グラフィックアプリケーションを実行するための強力なハードウェアに膨大なコストがかかります。また、FTPやメールなどの簡易的な手段で大きなサイズのデザインファイルを共有するなど、データ管理に関する問題も抱えています。

XenDesktopで物理デスクトップをホストする配信方法を使用すると、単一のデスクトップイメージを複数のワークステーションやブレードサーバーに提供でき、3Dグラフィックアプリケーションをネイティブのオペレーティングシステム上で実行するためのハイパーバイザーも不要です。

すべてのファイルはネットワーク内の中央データセンターに格納され、ワークステーション間で転送されることはありません。このため、ネットワーク内でデザインファイルをより高速かつ安全に共有できます。

大規模なコールセンターの運営には、ピーク時と非ピーク時に適切な人員やマシンを配するという困難な課題が伴います。

VDIをプールして配信する方法を使用すると、最小限のコストで多くのユーザーに標準化されたデスクトップへのアクセスを動的に提供できます。プールされた仮想マシンは、先にログオンしたユーザーから順に個別のセッションで割り当てられます。

セッション内のデスクトップで変更された内容はユーザーのログオフ時に破棄されるため、これらの仮想マシンに対する日々のメンテナンス作業が軽減されます。また、セキュリティも向上します。

コールセンターを効率化するもう1つの配信方法として、XenAppでデスクトップをホストすることができます（従来の「公開デスクトップ」）。この方法では、単一のサーバーオペレーティングシステムで複数のユーザーデスクトップをホストします。

VDIをプールする方法よりも低コストですが、ユーザーがアプリケーションをインストールしたり、システム設定を変更したり、サーバーを再起動したりすることは許可されません。

# 新しい展開環境

Jan 04, 2017

XenAppまたはXenDesktopの環境を構築するには

1. XenAppまたはXenDesktop環境のコンポーネントをホストおよび管理するための仮想化環境をセットアップします。サポートされる仮想化プラットフォーム、管理ツール、およびクラウド展開ソリューションについては、「[システム要件](#)」を参照してください。

管理者は、これらの仮想化プラットフォームを使用してXenAppやXenDesktopの環境をホストおよび管理できます。

- XenServer。XenServerのセットアップおよび使用については、「[XenServer](#)」を参照してください。
- VMware vSphere。XenAppまたはXenDesktopの環境でのVMware vSphereのセットアップおよび使用については、「[仮想化環境の準備：VMware](#)」を参照してください。
- Hyper-VとMicrosoft System Center Virtualization Machine Manager (VMM) XenAppまたはXenDesktopの環境でのHyper-V with VMMのセットアップおよび使用については、「[仮想化環境の準備：Microsoft System Center Virtual Machine Manager](#)」を参照してください。

Microsoft System Center Configuration ManagerとCitrix Connector 7.5 for System Center Configuration Manager 2012を使用すると、XenApp/XenDesktop環境の物理マシンおよび仮想マシンを管理したり、リモートPCアクセスでのWake on LAN機能を有効にしたりできます。「[Microsoft System Center Configuration Managerを使用するための準備](#)」を参照してください。

以下のクラウド展開ソリューションを使用して、製品コンポーネントをホストしたり仮想マシンをプロビジョニングしたりできます。これらのソリューションでは、コンピューティングリソースをプールしてパブリック、プライベート、およびハイブリッドのIaaS (Infrastructure as a Service) クラウドを構築できます。

- Amazon Web Services (AWS) — 「[Deploy XenApp and XenDesktop 7.5 and 7.6 with Amazon VPC](#)」を参照してください。
- Citrix CloudPlatform — 「[XenApp and XenDesktop concepts and deployment on CloudPlatform](#)」を参照してください。

2. XenApp/XenDesktopサイトの構築に必要な非Citrixインフラストラクチャコンポーネントをセットアップします。これには、Active Directoryドメインサービスを実行する少なくとも1つのドメインコントローラーが含まれます。
3. XenApp/XenDesktopサイトを構成するCitrixコンポーネントをインストールします。各コンポーネントは、インストールウィザードに表示される指示に従ってインストールしたり、コマンドラインインターフェイスでスクリプトを実行してインストールしたりできます。どちらの方法でも、この製品の実行に必要なサードパーティ製ソフトウェアの多くが自動的にインストールされます。
  1. インストールを始める前に、「[システム要件](#)」を確認してください。また、「[インストールの準備](#)」を読んで、必要なタスクを完了しておいてください。
  2. Delivery Controller、Citrix Studio、Citrix Director、Citrixライセンスサーバー、Citrix StoreFrontなどのコアコンポーネントをインストールします。コンポーネントのインストールについては、「[グラフィカルインターフェイスを使ったインストール](#)」または「[コマンドラインを使ったインストール](#)」を参照してください。
  3. Studioでサイトを作成します。詳しくは、「[サイトの作成](#)」を参照してください。
  4. Virtual Delivery Agent (VDA) をインストールします。仮想マシンの作成に使用するマスターイメージにインストールしたり、各マシン上に直接インストールしたりできます。VDAのインストールについては、「[グラフィカルインターフェイスを使ったインストール](#)」または「[コマンドラインを使ったインストール](#)」を参照してください。また、「[Active Directoryのスクリプトを使ったVirtual Delivery Agentのインストールまたは削除](#)」にも関連情報が記載されています。リモートPCアクセス機能を使用する場合は、オフィスにある各ユーザーのコンピューター上にVDA for Desktop OSをイ



インストールします。コマンドラインインターフェイスや既存の電子ソフトウェア配信 (ESD) ツールを使用してVDAをインストールすることをお勧めします。

5. 必要に応じて、プリントサーバー上にUniversal Print Serverをインストールします。Universal Print Serverのインストールについては、「[グラフィカルインターフェイスを使ったインストール](#)」または「[コマンドラインを使ってインストール](#)」を参照してください。
4. 必要に応じて、追加のCitrixコンポーネントをXenApp/XenDesktop展開に統合します。次に例を示します。
  - Provisioning ServicesはXenAppおよびXenDesktopのオプションコンポーネントで、マスターイメージをターゲットデバイスにストリーム配信してマシンをプロビジョニングします。「[Provisioning Services](#)」を参照してください。
  - Citrix NetScaler Gatewayはアプリケーションアクセスのセキュリティを保護するソリューションで、詳細なアプリケーションレベルのポリシーと操作の制御機能を管理者に提供し、アプリケーションとデータへのアクセスのセキュリティを保護します。「[Citrix NetScaler Gateway](#)」を参照してください。
  - Citrix CloudBridgeは、WAN接続のパフォーマンスを最適化するための一連のアプライアンスです。「[Citrix CloudBridge](#)」を参照してください。
5. ユーザーに配信するリソースをセットアップします。セットアップ方法は、使用する配信方法により異なります。基本的には、以下の手順を行います。
  1. ハイパーバイザー側の管理ツールを使用して、配信するデスクトップやアプリケーションを定義するマスターイメージを作成します。「[マスターイメージの準備](#)」を参照してください。
  2. マスターイメージからプロビジョニングされる物理マシンや仮想マシンのマシンカタログを作成します。「[マシンカタログの作成](#)」を参照してください。
    - Machine Creation Servicesを使用してマシンをプロビジョニングする場合は、Studioでマシンをマシンカタログに追加できます。
    - Provisioning Servicesを使用してマシンをプロビジョニングする場合は、Provisioning Servicesコンソールでマシンをマシンカタログに追加します。
  3. Studioでデリバリーグループを作成して、これらのマシンおよびマシン上にインストールされたアプリケーションにアクセスできるユーザーを指定します。「[デリバリーグループ](#)」を参照してください。

# インストールの準備

Jan 04, 2017

次の表には、コアコンポーネント（Delivery Controller、Citrix Studio、Citrix Director、Citrixライセンスサーバー、StoreFront）およびVirtual Delivery Agent（VDA）をインストールする前に完了させるべきタスクや考慮または注意が必要な事項について説明されています。

| • | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | <p>確認事項：</p> <ul style="list-style-type: none"><li>この製品を初めて使用する場合は、「<a href="#">Technical overview</a>」および関連トピックを参照してください。</li><li>「<br/>— <a href="#">既知の問題</a><br/>」を参照して、インストールに関する既知の問題について確認してください。</li><li>クラウド環境でのインストールについては、次を参照してください。<ul style="list-style-type: none"><li>Amazon Web Services (AWS) — 「<a href="#">Deploy XenApp and XenDesktop 7.5 and 7.6 with Amazon VPC</a>」</li><li>Citrix CloudPlatform — 「<a href="#">XenAppとXenDesktopの概念とCloudPlatformでの展開</a>」</li></ul></li><li>仮想化環境としてXenServerを使用する場合は、XenServerのドキュメントを参照してください。</li><li>仮想化環境としてVMwareまたは<a href="#">Microsoft System Center Virtual Machine Manager</a>を使用する場合は、それぞれリンク先のドキュメントを参照してください。</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|   | <p>各コンポーネントをどこにインストールするか決定し、次にマシンおよびオペレーティングシステムを準備します。</p> <ul style="list-style-type: none"><li>Controller、Studio、Director、仮想化リソース、Virtual Delivery Agentでサポートされるオペレーティングシステムおよびそのバージョンについては、「<a href="#">システム要件</a>」を参照してください。Citrix StoreFrontおよびCitrixライセンスサーバーの要件に関するドキュメントでは、サポートされるプラットフォームが指定されています。<ul style="list-style-type: none"><li>コアコンポーネントは、同じサーバー上にインストールしたり別のサーバー上にインストールしたりできます。たとえば、小規模な展開をリモートで管理する場合、Controllerをインストールしたサーバーとは異なるマシン上にStudioをインストールできます。将来的にサイトを拡張する計画がある場合は、各コンポーネントを個別のサーバー上にインストールします。たとえば、ライセンスサーバーとDirectorを別のサーバー上にインストールします。</li><li>Delivery ControllerとVirtual Delivery Agent for Windows Server OSを同一サーバー上にインストールできます。インストーラーを起動してDelivery Controller（およびインストールするその他のコアコンポーネント）をインストールし、インストーラーを再度起動してVirtual Delivery Agent for Windows Server OSをインストールします。</li><li>XenApp 6.5 Feature Pack 2 for Windows Server 2008 R2またはそれ以前のバージョンのXenAppが動作するサーバーにはStudioをインストールしないでください。</li></ul></li><li>各オペレーティングシステムを最新の状態にアップデートしておく必要があります。</li><li>すべてのマシンのシステムクロックを同期しておく必要があります。これは、Kerberosでマシン間の通信を保護するために必要です。</li><li>デフォルトでは、C:\Program Files\Citrixに各コンポーネントがインストールされます。インストール時に別の場所を指定することもできますが、Network Serviceアカウントでの実行権限が必要です。</li><li>コンポーネントの必須条件は自動的にインストールされます。ただし、例外については、「<br/>— <a href="#">システム要件</a><br/>」を確認してください。</li></ul> |
|   | <p>サイト構成データベース用のSQL Serverソフトウェアをインストールする場所を決定します。</p> <ul style="list-style-type: none"><li>デフォルトでは、Controllerのインストール時にSQL Server 2012 Expressが必要に応じて自動的にインストールされます。</li></ul> <p>デフォルトのインストールでは、デフォルトのWindowsサービスアカウントおよび権限を使用します。Windowsサービスアカウントをsysadminロールに追加する方法など、デフォルトの設定について詳しくは、Microsoft社のドキュメントを参照してください。Controllerは、この構成でNetwork Serviceアカウントを使用します。SQL Serverに追加のロールまたは権限は必要ありません。</p> <p>必要に応じて、データベースインスタンスで[インスタンスの非表示]を選択できます。Studioでデータベースのアドレスを構成する場合、インスタンス名ではなく、インスタンスの静的ポート番号を入力してください。SQL Serverデータベースエンジンのインスタンスを非表示にする方法について詳しくは、Microsoft社のドキュメントを参照してください。</p> <ul style="list-style-type: none"><li>また、適切なバージョンのSQL Serverをそのサーバーに別途インストールしたり、別のサーバーにインストールしたりできます。この場合、コアコンポーネントのインストール時にSQL Serverをインストールする必要はありません。ただし、サイトを作成する前にSQL Serverをインストールしておく必</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>• 説明<br/>       要があります。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <p>— 計画<br/>       」を参照してデータベースに関する考慮事項を確認し、サポートされている任意の冗長性インフラストラクチャをセットアップします。</p> <p>重要：Controllerとデータベース間の接続にはWindows認証が必要です。</p>                                                                                                                                                                                                                                                                                                                                                                                                   |
| <p>通信ポートをどのように開放するかを決定します。</p> <p>Windowsファイアウォールサービスが実行されている場合、ファイアウォールが無効になっていてもデフォルトでこれらのポートが開放されます。サードパーティ製のファイアウォールを使用している場合やファイアウォールを使用していない場合、またはインストール完了後に手作業でポートを開放する場合は、このデフォルトの動作を無効にすることができます。</p> <ul style="list-style-type: none"> <li>• Controller：TCP 80、443</li> <li>• Director：TCP 80、443</li> <li>• ライセンスサーバー：TCP 7279、8082、8083、27000</li> <li>• StoreFront：TCP 80、443</li> </ul> <p>ヒント：ポート情報の詳細については、<a href="#">CTX101810</a> を参照してください。そのほかのインストールオプションについては、「<a href="#">コマンドラインを使ったインストール</a>」を参照してください。</p> |
| <p>Active Directoryドメインを構成します。</p> <ul style="list-style-type: none"> <li>• コアコンポーネントをインストールするには、ドメインユーザーであることに加えて、そのマシンのローカル管理者である必要があります。</li> <li>• ドメインコントローラーとして動作するコンピューター上にはいずれのコンポーネントもインストールしないでください。</li> <li>• サポートされる機能レベルについては、「<br/>       — システム要件<br/>       」を参照してください。手順については、Microsoft社のドキュメントを参照してください。</li> </ul> <p>ライセンスサーバーをインストールする管理者は、そのライセンスサーバーのすべての管理タスクの実行権限を持つ管理者として設定されます。</p>                                                                                                                         |
| <p>Directorをインストールする前に、Directorのシャドウ機能を使用するかどうかを指定します。この機能では、Windowsリモートアシスタンスが使用されません。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <p>ヒント：</p> <ul style="list-style-type: none"> <li>• 正しくインストールできないコンポーネントがあった場合は、処理が停止してエラーメッセージが表示されます。この時点でインストール済みのコンポーネントは、インストールし直す必要はありません。</li> <li>• Studioは、インストールの完了時に自動的に起動します。インストール時にこの動作を無効にすることもできます。</li> <li>• インストールの前、最中、および後に作成するオブジェクト（ネットワーク、マシンカタログ、デリバリーグループ、リソースなど）には、重複しない名前を指定してください。</li> <li>• Amazon Web Services (AWS) でコンポーネントをインストールしたら、Studioでサイトを作成するときにリージョン、アベイラビリティゾーン、VPC名、サブネットアドレス、ドメイン名、セキュリティグループ名、および資格情報が必要になります。</li> </ul>                                                               |

|                                                                                                                                                                                                              |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>✔ 説明</p>                                                                                                                                                                                                  |
| <p>VDA for Windows Desktop OSをインストールする場合は、HDX 3D Proバージョンをインストールするか通常のバージョンをインストールするかを指定します。</p> <p>HDX 3D Pro機能を使用すると、GPUでのハードウェアアクセラレーションに適したデスクトップおよびアプリケーションを配信できます。詳しくは、HDX 3D Proのドキュメントを参照してください。</p> |
| <p>VDAの使用方法を決定します。</p> <p>デフォルトの設定では、VDAがインストールされたマスターイメージをMachine Creation ServicesまたはProvisioning Servicesで使用してほかの仮想マシンを作成することが想定されています。VDAを既存のマシン上にインストールする場合は、このデフォルトの設定を無効にできます。</p>                    |

説明

Citrix Receiver for Windows (CitrixReceiver.exe) をインストールするかどうかを指定します。

このデフォルトの動作は無効にできます。

通信ポートをどのように開放するかを決定します。

Windowsファイアウォールサービスが実行されている場合、ファイアウォールが無効になっていてもデフォルトでこれらのポートが開放されます。 サードパーティ製のファイアウォールを使用している場合やファイアウォールを使用していない場合、またはインストール完了後に手作業でポートを開放する場合は、このデフォルトの動作を無効にすることができます。

- Controller : TCP 80、1494、2598、8008
  - ユーザーデバイスと仮想デスクトップ間の通信については、ポート1494と2598上の受信TCPトラフィックをポート例外として構成します。 セキュリティ上の理由から、これらの登録ポートをICAプロトコルとCommon Gateway Protocol以外の用途で使用しないことをお勧めします。
  - Controllerと仮想デスクトップ間の通信については、ポート80上の受信トラフィックをポート例外として構成します。
- Windowsリモートアシスタンス : TCP 3389  
この機能が有効な場合、このポートはWindowsにより自動的に開放されます (ポートを手作業で開放することを選択した場合を含む)。
- リアルタイムオーディオ転送 : UDP 16500~16509

ヒント : Citrix製品で使用されるポートについては、[CTX101810](#)を参照してください。

インストール済みのControllerの指定方法を決定します。


- 手動で指定する : Controllerの完全修飾ドメイン名 (FQDN) を入力します。 ドメインに属していないControllerを指定することもできますが、VDAで接続できるのはドメインに属しているControllerのみです。 また、ドメインに属していないControllerの接続をテストすることはできません。
- Active Directoryから場所を選択する : コントローラーがドメインに属している場合は、Active Directoryを使用します。
- Machine Creation Servicesで自動的に指定する : Machine Creation ServicesによりControllerが指定されます。
- 後で指定する : 後でこのインストーラーを再度実行したり、Citrixポリシーやレジストリ値を構成したり、Active Directory OUを使用したりして指定します。

CitrixグループポリシーでControllerの場所を指定する場合、インストール時の指定が無視されます。

Controllerの場所を指定した後でほかのControllerを追加した場合は、自動更新機能を使用してVDAを更新できます。

以下の機能を使用するかどうかを決定します。

- パフォーマンスの最適化。この機能を有効にすると、ハイパーバイザー上の仮想マシンにインストールされるVDAが最適化されます。仮想マシンの最適化には、オフラインファイルの無効化、バックグラウンド最適化 (デフラグ処理) の無効化、およびイベントログサイズの縮小などの操作が含まれます。詳しくは、[CTX125874](#)を参照してください。 リモートPCアクセスを使用する場合は、このオプションを有効にしないでください。デフォルトでは有効になっています。
- Windowsリモートアシスタンス。この機能を有効にすると、Directorのシャドウ機能でWindowsリモートアシスタンスが使用されます。また、ファイアウォールのTCPポート3389が自動的に開放されます (ファイアウォールポートを手動で構成することを選択してもこのポートは開放されます)。デフォルトでは有効になっています。

|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p><b>説明</b> アルタイムオーディオ転送。この機能を有効にすると、オーディオパケットにUDPが使用され、オーディオパフォーマンスが向上します。デフォルトでは有効になっています。</p> <ul style="list-style-type: none"> <li>Personal vDisk（仮想マシンにVDA for Windows Desktop OSをインストールする場合にのみ選択可能）。この機能を有効にすると、マスターイメージでPersonal vDiskが使用されます。詳しくは、「<a href="#">Personal vDisk</a>」を参照してください。デフォルトでは無効になっています。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                                                                                   | <p>ヒント：</p> <ul style="list-style-type: none"> <li>Windowsサーバーでは、印刷スプーラーサービスがデフォルトで有効になります。このサービスを無効にすると、VDA for Windows Server OSをインストールできなくなります。このため、VDAのインストール前にこのサービスが有効であることを確認してください。</li> <li>インストール時にオペレーティングシステムが自動的に認識されるため、適切なVDA（VDA for Windows Server OSまたはVDA for Windows Desktop OS）がインストールされます。</li> <li>VDAのインストール時に、Profile managementがインストールされます。</li> <li>VDAをインストールすると、Direct Access Usersと呼ばれる新しいローカルユーザーグループが自動的に作成されます。このユーザーグループは、VDA for Windows Desktop OSではRDP接続でのみ使用され、VDA for Windows Server OSではICA接続およびRDP接続で使用されます。</li> <li>VDA for Windows Server OSをインストールすると、リモートデスクトップサービスの役割サービスが自動的にインストールされて有効になります。</li> <li>リモートPCアクセス構成の場合は、ユーザーがリモートでアクセスする社内の物理PC上にVDA for Windows Desktop OSをインストールします。</li> <li>VDAは、XenAppまたはXenDesktop製品のISOからインストールしたり、スタントアロンのVDAインストールパッケージを使用してインストールしたりできます。詳しくは、「<a href="#">スタンドアロンパッケージによるVDAのインストール</a>」を参照してください。</li> </ul> |

最新のVirtual Delivery Agent（VDA）はWindows XPまたはWindows Vistaシステムではサポートされません。また、新しい機能の一部もこれらのオペレーティングシステムではサポートされません。このリリースのすべての機能を使用できるように、Windows XPまたはWindows VistaのシステムをWindows 7、Windows 8、またはWindows 10に置き換えて、このリリースのVirtual Delivery Agentをインストールすることをお勧めします。

Windows XPまたはWindows Vistaを実行しているマシンを引き続き使用する必要がある場合は、古いバージョンのVirtual Desktop Agent（5.6 FP1およびいくつかのHotfix）をインストールできます。詳しくは、[CTX140941](#)を参照してください。

次の点に注意してください。

- Windows XPまたはWindows Vistaシステムには、コアコンポーネント（Controller、Studio、Director、StoreFront、Citrixライセンスサーバー）をインストールできません。
- リモートPCアクセスは、Windows Vistaをサポートしません。
- CitrixでのWindows XPのサポートは、Microsoft社によるサポートが終了した2014年4月8日に終了しました。
- 古いバージョンのVDAを継続使用すると、新しい機能やControllerへのVDA登録で問題が発生することがあります。詳しくは、「[混在環境の考慮事項](#)」を参照してください。

# 仮想化環境の準備：VMware

Jan 04, 2017

VMwareを使用して仮想マシンを提供する場合は、このトピックのガイダンスに従ってください。

1. vCenter Serverおよび必要な管理ツールをインストールします（vSphere vCenterのリンクモードはサポートされません）。
2. 次の権限があるVMwareユーザーアカウントを、DataCenterまたはそれ以上のレベルで作成します。このアカウントは、vCenterとの通信で使用されます。また、新しい仮想マシンを作成するための権限が必要です。

| SDK                                          | ユーザーインターフェイス                    |
|----------------------------------------------|---------------------------------|
| Datastore.AllocateSpace                      | [データストア] > [領域の割り当て]            |
| Datastore.Browse                             | [データストア] > [データストアの参照]          |
| Datastore.FileManagement                     | [データストア] > [低レベルのファイル操作]        |
| Network.Assign                               | [ネットワーク] > [ネットワークの割り当て]        |
| Resource.AssignVMToPool                      | [リソース] > [仮想マシンのリソースプールへの割り当て]  |
| System.Anonymous、System.Read、および System.View | 自動的に追加されます。                     |
| Task.Create                                  | [タスク] > [タスクの作成]                |
| VirtualMachine.Config.AddRemoveDevice        | [仮想マシン] > [構成] > [デバイスの追加または削除] |
| VirtualMachine.Config.AddExistingDisk        | [仮想マシン] > [構成] > [既存ディスクの追加]    |
| VirtualMachine.Config.AddNewDisk             | [仮想マシン] > [構成] > [新規ディスクの追加]    |
| VirtualMachine.Config.AdvancedConfig         | [仮想マシン] > [構成] > [詳細]           |
| VirtualMachine.Config.CPUCount               | [仮想マシン] > [構成] > [CPUカウントの変更]   |
| VirtualMachine.Config.Memory                 | [仮想マシン] > [構成] > [メモリ]          |

|                                              |                                                 |
|----------------------------------------------|-------------------------------------------------|
| VirtualMachine.Config.RemoveDisk<br>SDK      | [仮想マシン] > [構成] > [ディスクの削除]<br>ユーザーインターフェイス      |
| VirtualMachine.Config.Resource               | [仮想マシン] > [構成] > [リソースの変更]                      |
| VirtualMachine.Config.Settings               | [仮想マシン] > [構成] > [設定]                           |
| VirtualMachine.Interact.PowerOff             | [Virtual machine] > [Interaction] > [Power Off] |
| VirtualMachine.Interact.PowerOn              | [Virtual machine] > [Interaction] > [Power On]  |
| VirtualMachine.Interact.Reset                | [Virtual machine] > [Interaction] > [Reset]     |
| VirtualMachine.Interact.Suspend              | [Virtual machine] > [Interaction] > [Suspend]   |
| VirtualMachine.Inventory.Create              | [仮想マシン] > [インベントリ] > [新規作成]                     |
| VirtualMachine.Inventory.CreateFromExisting  | [仮想マシン] > [インベントリ] > [既存のものから作成]                |
| VirtualMachine.Inventory.Delete              | [仮想マシン] > [インベントリ] > [削除]                       |
| VirtualMachine.Inventory.Register            | [仮想マシン] > [インベントリ] > [登録]                       |
| VirtualMachine.Provisioning.Clone            | [仮想マシン] > [プロビジョニング] > [仮想マシンのクローン作製]           |
| VirtualMachine.Provisioning.DiskRandomAccess | [仮想マシン] > [プロビジョニング] > [ディスクアクセスの許可]            |
| VirtualMachine.Provisioning.GetVmFiles       | [仮想マシン] > [プロビジョニング] > [仮想マシンのダウンロードの許可]        |
| VirtualMachine.Provisioning.PutVmFiles       | [仮想マシン] > [プロビジョニング] > [仮想マシンファイルのアップロードの許可]    |
| VirtualMachine.Provisioning.DeployTemplate   | [仮想マシン] > [プロビジョニング] > [テンプレートの展開]              |
| VirtualMachine.Provisioning.MarkAsVM         | [仮想マシン] > [プロビジョニング] > [仮想マシンとしてマーク]            |

|                                            |                                                                                                                                          |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| VirtualMachine.State.CreateSnapshot<br>SDK | vSphere 5.0, Update 2およびvSphere 5.1, Update 1 : [仮想マシン] > [状態] > [スナップショットの作成]<br>ユーザーインターフェイス                                           |
|                                            | vSphere 5.5 : [仮想マシン] > [スナップショット管理] > [スナップショットの作成]                                                                                     |
| VirtualMachine.State.RemoveSnapshot        | vSphere 5.0, Update 2およびvSphere 5.1, Update 1 : [仮想マシン] > [状態] > [スナップショットの削除]<br>vSphere 5.5 : [仮想マシン] > [スナップショット管理] > [スナップショットの削除]   |
| VirtualMachine.State.RevertToSnapshot      | vSphere 5.0, Update 2およびvSphere 5.1, Update 1 : [仮想マシン] > [状態] > [スナップショットまで戻る]<br>vSphere 5.5 : [仮想マシン] > [スナップショット管理] > [スナップショットまで戻る] |

3. 作成する仮想マシンにタグを設定する場合は、ユーザーアカウントに次の権限も必要です。

| SDK                       | ユーザーインターフェイス          |
|---------------------------|-----------------------|
| Global.ManageCustomFields | [グローバル] > [カスタム属性の管理] |
| Global.SetCustomField     | [グローバル] > [カスタム属性の設定] |

クリーンな基本イメージで新しい仮想マシンを作成できるように、Machine Creation Servicesで作成された仮想マシンにタグを設定して、基本イメージとして使用する仮想マシンの一覧から除外してください。

vSphere通信を保護するため、HTTPではなくHTTPSを使用することをお勧めします。HTTPSを使用するにはデジタル証明書が必要です。組織のセキュリティポリシーに従って、証明書機関により発行されるデジタル証明書を使用することをお勧めします。

証明機関のデジタル証明書を使用できない場合は、VMwareによりインストールされる自己署名証明書を使用することもできます（組織のセキュリティポリシーで許可される場合）。VMware vCenterの証明書を各Controllerに追加します。次の手順に従います。

- vCenter Serverを実行しているコンピューターの完全修飾ドメイン名（FQDN）を、そのサーバーのホストファイル（%SystemRoot%/WINDOWS/system32/Drivers/etc/）に追加します。この手順は、vCenter Serverを実行しているコンピューターのFQDNがドメイン名システムに登録されていない場合にのみ必要です。
- 以下のいずれかの方法で、vCenterの証明書を入手します。
  - vCenterサーバーからコピーする。
    - vCenterサーバー上のrui.crtファイルを、Delivery Controllerからアクセス可能な場所にコピーします。
    - Controllerで、エクスポートした証明書の保存先に移動し、rui.crtファイルを開きます。
  - Webブラウザでダウンロードする。Internet Explorerで証明書をダウンロードするには、Internet Explorerを右クリックして「管理者として実行」を選択しなければならない場合があります。



1. Webブラウザを開き、vCenterサーバーへの保護された接続 (https://server1.domain1.comなど) を確立します。
  2. セキュリティに関する警告を受け入れます。
  3. 証明書のエラーが表示されるアドレスバーをクリックします。
  4. 証明書を表示して、[詳細] タブをクリックします。
  5. [ファイルへコピー] を選択して、任意のファイル名を指定してCER形式でエクスポートします。
  6. エクスポートした証明書を保存します。
  7. エクスポートした証明書のCERファイルを開きます。
- 管理者として実行するInternet Explorerで直接インポートする。
    1. Webブラウザを開き、vCenterサーバーへの保護された接続 (https://server1.domain1.comなど) を確立します。
    2. セキュリティに関する警告を受け入れます。
    3. 証明書のエラーが表示されるアドレスバーをクリックします。
    4. 証明書を表示します。
  - 各Controller上の証明書ストアに証明書をインポートします。
    1. [証明書のインストール] をクリックして [ローカルマシン] を選択し、[次へ] をクリックします。
    2. [証明書をすべて次のストアに配置する] を選択して、[参照] をクリックします。
    3. Windows Server 2008 R2の場合：
      1. [物理ストアを表示する] チェックボックスをオンにします。
      2. [信頼されたユーザー] を開きます。
      3. [ローカルコンピューター] を選択します。
      4. [次へ] をクリックし、[完了] をクリックします。
    - Windows Server 2012またはWindows Server 2012 R2の場合：
      1. [信頼されたユーザー] を選択し、[OK] をクリックします。
      2. [次へ] をクリックし、[完了] をクリックします。
- 重要：インストール後にvSphereサーバーの名前を変更する場合は、サーバー上で新しい自己署名証明書を作成してから、新しい証明書をインポートする必要があります。

管理者は、ユーザーのデスクトップおよびアプリケーションを提供するためのマスター仮想マシンを作成します。

1. マスター仮想マシンにVDAをインストールします。このとき、デスクトップを最適化するオプションを選択すると、パフォーマンスが向上します。
2. バックアップのため、マスター仮想マシンのスナップショットを作成します。詳しくは、「[マスターイメージの準備](#)」を参照してください。

既存のマシンカタログを選択する代わりにStudioで仮想マシンを作成する場合は、仮想デスクトップ作成用のホストインフラストラクチャをセットアップするときに以下の情報を指定します。

1. ホストの種類としてVMware vSphereを選択します。
2. vCenter SDKのアクセスポイントのアドレス (https://vmware.example.com/sdkなど) を入力します。
3. 新しい仮想マシンを作成する権限を持つ、既存のVMwareユーザーアカウントの資格情報を入力します。ユーザー名を「<domain/username>」形式で指定します。

# 仮想化環境の準備：Microsoft System Center Virtual Machine Manager

Jan 04, 2017

Hyper-VとMicrosoft System Center Virtual Machine Manager (VMM) を使用して仮想マシンを提供する場合は、このトピックのガイダンスに従ってください。

このリリースでは、以下のVMMがサポートされています。

- VMM 2012 — 管理機能が向上し、仮想マシンだけでなく仮想化されたデータセンター全体を管理できます。このリリースには、Windows Server Update Servicesとの統合に加えて、クラスターホストへの修正プログラムの適用をオーケストレーションする機能も用意されており、各ホストに必要な修正プログラムのレベルを定義できます。
- VMM 2012 SP1 — クラスター化された共有ボリュームおよびStorage Area Network (SAN) のファイルサーバー上でServer Message Block (SMB) 3.0を使用する場合のMachine Creation Services (MCS) のパフォーマンスが向上しました。これらのファイル共有では低コストのキャッシュ機能が提供され、SANストレージ上のI/Oが減少してパフォーマンスが向上します。
- VMM 2012 R2 — 実行中の仮想マシンのスナップショット作成、VHDXの動的なサイズ変更、および記憶域など、Windows Server 2012 R2の主要な機能をスケーラブルに管理できます。

このリリースは、VMM 2012 R2の第1世代の仮想マシンのみをサポートします。第2世代の仮想マシンは、Machine Creation Services (MCS) およびProvisioning Servicesでサポートされません。MCSまたはProvisioning Servicesを使って仮想マシンを作成する場合、マスター仮想マシンの一覧に第2世代の仮想マシンが表示されません。第2世代の仮想マシンはデフォルトでセキュアブートが有効になっているため、VDAが正しく機能しません。

- VMM 2012からVMM 2012 SP1またはVMM 2012 R2へのアップグレード  
VMMおよびHyper-Vホストの要件については、<http://technet.microsoft.com/ja-jp/library/gg610649.aspx>を参照してください。VMMコンソールの要件については、<http://technet.microsoft.com/ja-jp/library/gg610640.aspx>を参照してください。

Hyper-Vの混在クラスターはサポートされません。混在クラスターとは、Hyper-V 2008とHyper-V 2012が混在するものなどを指します。

- VMM 2008 R2からVMM 2012 SP1へのアップグレード  
VMM 2008 R2を使用するXenDesktop 5.6をアップグレードする場合は、XenDesktopのダウンタイムを避けるため、次の順序でアップグレードしてください。
  1. VMM 2008 R2をVMM 2012にアップグレードします（これにより、XenDesktop 5.6とVMM 2012が動作することになります）。
  2. XenDesktopをアップグレードします（これにより、アップグレードされたXenDesktopとVMM 2012が動作することになります）。
  3. VMM 2012をVMM 2012 SP1にアップグレードします（これにより、アップグレードされたXenDesktopとVMM 2012 SP1が動作することになります）。
- VMM 2012 SP1からVMM 2012 R2へのアップグレード  
VMM 2012 SP1を使用するXenDesktopまたはXenApp 7.xをアップグレードする場合は、ダウンタイムを避けるため、次の順序でアップグレードしてください。
  1. XenDesktopまたはXenAppをアップグレードします（これにより、アップグレードされたXenDesktopまたはXenAppとVMM 2012 SP1が動作することになります）。
  2. VMM 2012 SP1をVMM 2012 R2にアップグレードします（これにより、アップグレードされたXenDesktopまたはXenAppとVMM 2012 R2が動作することになります）。

1. ハイパーバイザーをインストールして構成します。
  1. サーバー上にMicrosoft Hyper-V ServerおよびVMMをインストールします。すべてのDelivery ControllerがVMMサーバーと同じフォレストに含まれている必要があります。
  2. すべてのControllerにSystem Center Virtual Machine Managerコンソールをインストールします。
  3. 次のアカウント情報を確認します。
    - Studioでホストを指定するために使用するアカウントは、VMM管理者またはそのHyper-VマシンのVMM委任管理者である必要があります。このアカウントにVMMの委任管理者の役割のみがある場合は、ホストの作成時にストレージデータがStudioの一覧に表示されません。

- Studio統合に使用されるユーザーアカウントは、仮想マシンのライフサイクル管理（仮想マシンの作成、更新、および削除など）を実行できるように、各Hyper-Vサーバー上のAdministratorsローカルセキュリティグループのメンバーでもある必要があります。

注：Hyper-Vが動作するサーバー上にControllerをインストールすることはサポートされていません。

2. マスター仮想マシンを作成します。
  1. マスター仮想マシンにVirtual Delivery Agentをインストールします。このとき、デスクトップを最適化するオプションを選択してください。これにより、パフォーマンスが向上します。
  2. バックアップのため、マスター仮想マシンのスナップショットを作成します。  
詳しくは、「[マスターイメージの準備](#)」を参照してください。
3. 仮想デスクトップを作成します。MCSを使って仮想マシンを作成している場合、サイトまたは接続を作成すると
  1. ホストの種類として [Microsoft仮想化] を選択します。
  2. アドレスとして、ホストサーバーの完全修飾ドメイン名を入力します。
  3. 新しいVMを作成する権限を持つ、先にセットアップした管理者アカウントのアカウント情報を入力します。
  4. [ホスト詳細] ダイアログボックスで、仮想マシンの作成時に使用するクラスターまたはスタンドアロンホストを選択します。  
重要：単一Hyper-Vホストによる展開でも、クラスターまたはスタンドアロンホストを参照して選択します。

SMB 3ファイル共有の仮想マシンストレージ上でMCSを使用して作成されたマシンカタログの場合、ControllerのHCL (Hypervisor Communications Library) からの呼び出しをSMBストレージに適切に接続できるよう、資格情報が以下の要件を満たしていることを確認する必要があります。

- VMMのユーザー資格情報には、SMBストレージに対する完全な読み取りおよび書き込みアクセス権限が必要です。
- 仮想マシンのライフサイクルイベント中のストレージ仮想ディスク操作では、Hyper-Vサーバーを介してVMMのユーザー資格情報が使用されません。

Windows Server 2012のHyper-VとVMM 2012 SP1でSMBをストレージとして使用する場合は、Controllerから各Hyper-VマシンへのCredSSP (Authentication Credential Security Support Provider) を有効にしてください。詳しくは、[CTX137465](#)を参照してください。

標準のPowerShell V3リモートセッションを使用すると、HCLはCredSSPを使ってHyper-Vマシンへの接続を開きます。この機能では、Kerberosで暗号化されたユーザーの資格情報がHyper-Vマシンに渡され、この資格情報（この場合はVMMユーザーの資格情報）を使用してリモートのHyper-Vマシン上のセッション内でPowerShellコマンドが実行されます。これにより、ストレージに対する通信コマンドが正しく動作します。

以下のタスクでは、HCLからHyper-Vマシンに送信されてSMB 3.0ストレージ上で動作するPowerShellスクリプトを使用します。

- **マスターイメージの統合** — マスターイメージにより、新しいMCSプロビジョニングスキーム（マシンカタログ）が作成されます。作成された新しいディスクから新しい仮想マシンを作成できるようにマスター仮想マシンを複製およびフラット化（および元のマスター仮想マシンとの依存関係を削除）します。

root\virtualization\v2名前空間でConvertVirtualHardDiskを実行します。

例：  
\$ims = Get-WmiObject -class \$class -namespace "root\virtualization\v2"; \$result = \$ims.ConvertVirtualHardDisk(\$diskName, \$vhdstext) \$result

- **差分ディスクの作成** — マスターイメージを統合して作成されたマスターイメージから差分ディスクを作成します。この差分ディスクは、新しい仮想マシンに接続されます。

root\virtualization\v2名前空間でCreateVirtualHardDiskを実行します。

例：  
\$ims = Get-WmiObject -class \$class -namespace "root\virtualization\v2"; \$result = \$ims.CreateVirtualHardDisk(\$vhdstext); \$result

- **IDディスクのアップロード** — HCLでは、IDディスクをSMBストレージに直接アップロードすることはできません。そのため、Hyper-VマシンがIDディスクをストレージにアップロードしてコピーする必要があります。Hyper-VマシンはControllerからディスクを読み取れないため、HCLはHyper-Vマシンを介してIDディスクをコピーしておく必要があります。

1. HCLは管理者共有を介してIDディスクをHyper-Vマシンにアップロードします。
2. PowerShellリモートセッションで実行されるPowerShellスクリプトにより、Hyper-VマシンがIDディスクをSMBストレージにコピーします。Hyper-Vマシン上にフォルダーが作成され、（リモートPowerShell接続を介して）そのフォルダーに対する権限がVMMユーザーのみにロックされます。
3. HCLが管理者共有からファイルを削除します。
4. HCLがHyper-VマシンへのIDディスクのアップロードを完了すると、リモートPowerShellセッションによってIDディスクはSMBストレージにコピーされ、Hyper-Vマシンから削除されます。

IDディスクフォルダーが削除された場合は再作成され、再使用できるようになります。

- **IDディスクのダウンロード** — アップロードと同様に、IDディスクがHyper-VマシンからHCLに渡されます。次の処理により、Hyper-Vサーバー上にVMMユーザー権限のみを持つフォルダーが作成されます（存在しない場合）。
  1. PowerShell V3リモートセッションで実行されるPowerShellスクリプトにより、Hyper-VマシンがSMBストレージからローカルのHyper-VストレージにIDディスクをコピーします。
  2. HCLがHyper-Vマシンの管理者共有からIDディスクをメモリ内に読み取ります。
  3. HCLが管理者共有からファイルを削除します。

- **Personal vDiskの作成** — 管理者がPersonal vDiskマシンカタログで仮想マシンを作成する場合、空のディスク（Personal vDisk）を作成する必要があります。

空のディスクを作成する呼び出しでは、ストレージへの直接アクセスが不要です。メインまたはオペレーティングシステムディスクとは異なるストレージ上にPersonal vDiskがある場合は、リモートPowerShellを使って作成元の仮想マシンと同じ名前のディレクトリにPersonal vDiskを作成します。CSVまたはLocalStorageに対しては、リモートPowerShellを使用しないでください。空のディスクを作成する前にディレクトリを作成しておくと、VMMコマンドエラーを避けることができます。

Hyper-Vマシンから、ストレージ上でmkdirを実行します。

# Microsoft System Center Configuration Managerを使用するための準備

Jan 04, 2017

System Center Configuration Manager (Configuration Manager) 2012で物理デバイス上のアプリケーションやデスクトップへのアクセスを管理しているサイトでは、Configuration Managerの管理機能をXenAppやXenDesktop環境まで拡張できます。以下の統合オプションを使用できます。

- **Citrix Connector 7.5 for Configuration Manager 2012** — Citrix Connectorより、Configuration ManagerとXenApp/XenDesktopが連係して動作するようになります。Connectorを使用すると、Configuration Managerによる物理環境とXenApp/XenDesktopによる仮想環境の両方の保守・管理操作を統一させることができます。Connectorについては、「[Citrix Connector 7.5 for System Center Configuration Manager 2012](#)」を参照してください。
- **Configuration Managerのウェイクアッププロキシ機能** — Citrix Connectorを使用するかどうかにかかわらず、リモートPCアクセスのWake on LAN機能を使用するにはConfiguration Managerが必要です。詳しくは、「[Configuration ManagerとリモートPCアクセスのWake on LAN](#)」を参照してください。
- **XenApp/XenDesktopのプロパティ** — XenAppおよびXenDesktopのプロパティ設定により、Configuration Managerで管理するCitrix仮想デスクトップを識別できるようになります。これらのプロパティはCitrix Connectorにより自動的に設定されますが、以下で説明するように手作業での構成も可能です。

Microsoft System Center Configuration Manager 2012および2012 R2では仮想デスクトップを管理するためのプロパティを利用できます。

Configuration Manager 2012に表示されるプロパティのブール値は、trueとfalseではなく1と0で表示されることがあります。

これらのプロパティは、Root\Citrix\DesktopInformation名前空間のCitrix\_virtualDesktopInfoクラスで使用できます。これらのプロパティの名前は、Windows Management Instrumentation (WMI) プロバイダーでのものです。

| プロパティ              | 説明                                                                                                                                                                          |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AssignmentType     | IsAssignedの値を設定します。以下の値を指定します。 <ul style="list-style-type: none"><li>● ClientIP</li><li>● ClientName</li><li>● None</li><li>● User — これによりIsAssignedがTrueに設定されます。</li></ul> |
| BrokerSiteName     | サイト名です。HostIdentifierと同じ値を返します。                                                                                                                                             |
| DesktopCatalogName | デスクトップに関連付けられたマシンカタログの名前です。                                                                                                                                                 |
| DesktopGroupName   | デスクトップに関連付けられたデリバリーグループの名前です。                                                                                                                                               |
| HostIdentifier     | サイト名です。BrokerSiteNameと同じ値を返します。                                                                                                                                             |
|                    |                                                                                                                                                                             |

|                                                                               |                                                                                                                                                                                                                                                                                                 |
|-------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IsAssigned<br>プロパティ                                                           | デスクトップを各ユーザーに割り当てる場合はTrue、ランダムデスクトップの場合はFalseを説明を設定します。                                                                                                                                                                                                                                         |
| IsMasterImage                                                                 | <p>マスターイメージかどうかを指定します。たとえば、すべてのマシンがクリーンな状態で起動するように、プロビジョニングされたマシン上ではなくマスターイメージ上にアプリケーションをインストールできます。以下の値を指定します。</p> <ul style="list-style-type: none"> <li>• マスターイメージとして使用される仮想マシンではTrueになります（この値は、選択オプションに基づいてインストール時に設定されます）。</li> <li>• マスターイメージからプロビジョニングされる仮想マシンではClearedになります。</li> </ul> |
| IsVirtualMachine                                                              | 仮想マシンではtrue、物理マシンではfalseになります。                                                                                                                                                                                                                                                                  |
| OSChangesPersist                                                              | 再起動時にデスクトップのオペレーティングシステムイメージをクリーンな状態にリセットする場合はfalse、リセットしない場合はtrueになります。                                                                                                                                                                                                                        |
| PersistentDataLocation                                                        | Configuration Managerが永続データを格納する場所です。ユーザーはアクセスできません。                                                                                                                                                                                                                                            |
| PersonalDiskDriveLetter                                                       | デスクトップでPersonal vDiskを使用する場合に、そのPersonal vDiskに割り当てるドライブ文字です。                                                                                                                                                                                                                                   |
| BrokerSiteName、<br>DesktopCatalogName、<br>DesktopGroupName、<br>HostIdentifier | デスクトップのコントローラーへの登録時に設定されるため、未登録のデスクトップではNullになります。                                                                                                                                                                                                                                              |

これらのプロパティを収集するには、Configuration Managerでハードウェアインベントリを実行します。プロパティを表示するには、Configuration Managerのリソースエクスプローラーを使用します。これらのインスタンスでは、名前にスペースが含まれたり、プロパティ名とわずかに違ったものになったりすることがあります。たとえば、**BrokerSiteName**は、「Broker Site Name」と表示されることがあります。以下のタスクについて詳しくは、「[Citrix WMI Properties and System Center Configuration Manager 2012](#)」を参照してください。

- Configuration Managerを構成してCitrix VDAからCitrix WMIプロパティを収集する。
- Citrix WMIプロパティを使用してクエリベースのデバイスコレクションを作成する。
- Citrix WMIプロパティに基づいてグローバル条件を作成する。
- グローバル条件を使用してアプリケーションの展開の種類の要件を定義する。

また、MicrosoftクラスのCCM\_DesktopMachineのMicrosoftプロパティをRoot\ccm\_vdi名前空間で使用することもできます。詳しくは、Microsoft社のドキュメントを参照してください。

リモートPCアクセスの計画および展開については、「[リモートPCアクセス](#)」および「[ユーザーへのリモートPCアクセスの提供](#)」を参照してください。

リモートPCアクセスのWake on LAN機能を構成するには、以下のタスクを完了してから社内PCにVDAをインストールし、StudioでリモートPCアクセス展開を作成または編集します。

- 組織内にConfiguration Manager 2012を構成してからリモートPCアクセス用のすべてのマシンにConfiguration Managerクライアントを展開し、スケジュールされているSCCMインベントリサイクルが実行されるのを待ちます（必要に応じて、手動で強制的に実行することもできます）。StudioでConfiguration Manager接続を構成するときのアクセス資格情報には、セキュリティスコープにコレクションが含まれており、リモートツールオペレーターロールが付与されている必要があります。
- Intel社のActive Management Technology (AMT) を使用する場合。
  - PCでAMT 3.2.1以降がサポートされている必要があります。
  - 適切な資格情報およびプロビジョニングプロセスを使用して、PCでAMTが使用されるようにプロビジョニングします。
- Configuration Managerのウェイクアッププロキシやマジックパケットを使用する場合。
  - 各PCのBIOS設定で、Wake on LAN機能を有効にします。
  - Configuration Managerのウェイクアッププロキシの場合は、Configuration Managerでウェイクアッププロキシを有効にします。リモートPCアクセスのWake on LAN機能を使用するPCが属する各サブネットで、センチネルマシンとして動作可能なものが3台以上あることを確認します。
  - マジックパケットの場合は、サブネット宛てのブロードキャストまたはユニキャストを使用して、ネットワーク経路およびファイアウォールでパケットの転送がブロックされないようにします。

社内PC上にVDAをインストールしたら、StudioでリモートPCアクセス展開を作成するときに電源管理機能を有効または無効にします。

- 電源管理機能を有効にする場合は、接続の詳細としてConfiguration Managerのアドレス、アクセス資格情報、および名前を指定します。
- 電源管理機能を無効にした場合でも、電源管理 (Configuration Manager) 接続を後から追加して、リモートPCアクセスのマシンカタログを編集してこの接続を指定し、電源管理機能を有効にできます。

電源管理接続を編集すると、Configuration Managerのウェイクアッププロキシやマジックパケットの使用を構成したり、パケットの転送方法を変更したりすることができます。

# グラフィカルインターフェイスを使ったインストール

Jan 04, 2017

インストールを始める前に、「[インストールの準備](#)」を参照して必要なタスクを完了しておいてください。

次の手順に従って、インストーラーのグラフィカルインターフェイスを起動します。

1. 製品パッケージをダウンロードして圧縮を解除します。必要な場合は、ISOファイルからDVDを作成します。
2. ローカルの管理者アカウントを使って、インストール先のサーバーにログオンします。
3. DVDをドライブに挿入するか、ISOファイルをマウントします。インストーラーが自動的に起動しない場合は、AutoSelectファイルまたはマウントしたドライブをダブルクリックします。
4. インストールするコンポーネントを選択します。
  - 製品を初めてインストールする場合は、[Delivery Controller] をクリックします。これにより、Delivery Controllerがインストールされ、またオプションとしてStudio、Director、ライセンスサーバー、およびStoreFrontが同じサーバー上にインストールされます。
  - いくつかのコンポーネントがインストール済みの環境で展開を拡張する場合は、インストールするコンポーネントを [拡張展開] セクションで個別にクリックできます。このセクションでは、各コアコンポーネントと、Universal Print Server (プリントサーバー上にインストールします) を選択できます。
  - Virtual Delivery Agent (VDA) をインストールするには、選択可能なVDAエントリをクリックします。実行しているオペレーティングシステムに適したエントリのみが選択可能になっています。

インストールした後に、VDAをカスタマイズすることもできます。

1. プログラムを削除または変更するWindowsのコントロールパネルで[Citrix Virtual Delivery Agent <version-number>] を右クリックして、[変更] を選択します。
2. [Virtual Delivery Agent設定のカスタマイズ] を選択します。インストーラーが起動して、ControllerのアドレスやControllerへの登録時のTCP/IPポート (デフォルトは80) を変更したり、Windowsファイアウォールで自動的にポートを開くかどうかを変更したりできます。

グラフィックインターフェイスを使ってコンポーネントをアップグレードすることもできます。詳しくは、[環境のアップグレード](#)」を参照してください。

VDAは、XenAppまたはXenDesktop製品のISOからインストールしたり、スタントアロンのVDAインストールパッケージを使用してインストールしたりできます。詳しくは、「[スタンドアロンパッケージによるVDAのインストール](#)」を参照してください。

XenAppおよびXenDesktop 7.6 FP3では、UPSパッケージに更新バージョンのスタンドアロンUPSクライアントとサーバーコンポーネントが含まれています。インストール手順については、「[プリンターのプロビジョニング](#)」の「Citrix Universal Print Server (UPS) のインストール」を参照してください。



# コマンドラインを使ったインストール

Jan 04, 2017

コマンドラインインターフェイスでは、以下のインストールが可能です。

- Delivery Controller、Citrix Studio、Citrix Director、ライセンスサーバー、StoreFrontの1つまたは複数のコンポーネントをインストールする。
- マスターイメージ、仮想マシン、または物理マシン上にVirtual Delivery Agent (VDA) をインストールする。  
また、メディアに収録されているスクリプトをカスタマイズして、Active DirectoryでVDAをインストールしたり削除したりできます。

- インストール済みのVDAをカスタマイズする。
- ネットワークセッションプリンターをプロビジョニングするUniversal Print Serverをインストールする。ControllerにはUniversal Print Serverの機能が既に統合されているため、印刷サーバー上のみインストールします。

また、以下の管理タスクを実行できます。

- インストール済みのコンポーネントをこのバージョンから削除する。これを行うには、/removeオプションまたは/removeallオプションを使用します。詳しくは、「[コンポーネントの削除](#)」を参照してください。
- コンポーネントをアップグレードする。詳しくは、「[環境のアップグレード](#)」を参照してください。

コマンドの実行状態を確認して値を返すには、マシンの管理者であるか [管理者として実行] を使用する必要があります。詳しくは、Microsoft社のコマンドに関するドキュメントを参照してください。

重要：インストールを始める前に、「[インストールの準備](#)」を参照して必要なタスクを完了しておいてください。

製品メディアの\x64\XenDesktop Setupディレクトリから、XenDesktopServerSetup.exeコマンドを実行します。次の表は、このコマンドのオプションの一覧です。

注：XenAppをインストールするには、コマンドラインに/xenappオプションを追加します。XenDesktopをインストールする場合は、/xenappオプションを追加しないでください。

| オプション                                      | 説明                                                                                                                                                                                                                                                                                                                                                        |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /helpまたは/h                                 | コマンドのヘルプを表示します。                                                                                                                                                                                                                                                                                                                                           |
| /quietまたは/passive                          | ユーザーインターフェイスを表示せずにインストールを実行します。インストールプロセスは、Windowsタスクマネージャーにのみ表示されます。このオプションを指定しない場合、インストールウィザードが表示されます。                                                                                                                                                                                                                                                  |
| /logpath <path>                            | ログファイルのパスを指定します。既存のフォルダーを指定する必要があります（自動的に作成されません）。デフォルト値：“%TEMP%\Citrix\XenDesktop Installer”                                                                                                                                                                                                                                                             |
| /noreboot                                  | インストール後の再起動を無効にします（ほとんどのコアコンポーネントでは、デフォルトで再起動が無効になっています）。                                                                                                                                                                                                                                                                                                 |
| /remove                                    | /componentsオプションで指定したコアコンポーネントを削除します。                                                                                                                                                                                                                                                                                                                     |
| /removeall                                 | インストール済みのすべてのコアコンポーネントを削除します。                                                                                                                                                                                                                                                                                                                             |
| /xenapp                                    | XenAppをインストールします。このオプションを指定しない場合、XenDesktopがインストールされます。                                                                                                                                                                                                                                                                                                   |
| /configure_firewall                        | Windowsファイアウォールサービスが実行されている場合に（ファイアウォールが無効になっていても）、インストールされるコンポーネントで必要なポートが開放されます。サードパーティ製のファイアウォールを使用している場合は、適切なポートを手動で開く必要があります。                                                                                                                                                                                                                        |
| /components <component> [, <component>]... | 必須です。インストールまたは削除するコンポーネントをコンマ区切りのリストで指定します。以下の値を指定します。 <ul style="list-style-type: none"><li>• CONTROLLER - Controller</li><li>• DESKTOPSTUDIO - Studio</li><li>• DESKTOPDIRECTOR - Director</li><li>• LICENSESERVER - Citrixライセンスサーバー</li><li>• STOREFRONT - StoreFront</li></ul> このオプションを指定しない場合、すべてのコンポーネントがインストール（または、/removeオプションも指定されている場合は削除）されます。 |
| /installdir <directory>                    | コンポーネントのインストール先として既存の空ディレクトリを指定します。デフォルト値：c:\Program Files\Citrix                                                                                                                                                                                                                                                                                         |
| /tempdir <directory>                       | インストール時に一時ファイルを作成するディレクトリを指定します。デフォルト値：c:\Windows\Temp                                                                                                                                                                                                                                                                                                    |
| /nosql                                     | Controllerのインストール先サーバーにMicrosoft SQL Server Expressをインストールしない場合に指定します。このオプションを指定しない場合、SQL Server Expressがインストールされます。                                                                                                                                                                                                                                      |
| /no_remote_assistance                      | （Directorをインストールする場合にのみ有効）Windowsリモートアシスタンス機能をインストールせず、この機能を有効にしない場合に指定します。                                                                                                                                                                                                                                                                               |

たとえば、次のコマンドを実行すると、XenDesktop、Controller、Studio、Citrixライセンスサーバー、およびSQL Server Expressがサーバー上にインストールされます。コンポーネントの通信で使われるポートは自動的に開放されます。

```
\x64\XenDesktop Setup\XenDesktopServerSetup.exe /components controller,desktopstudio,licenseserver /configure_firewall
```

次のコマンドを実行すると、XenApp、Controller、Studio、およびSQL Server Expressがサーバー上にインストールされます。コンポーネントの通信で使用されるポートは自動的に開放されます。

```
\x64\XenDesktop Setup\XenDesktopServerSetup.exe /xenapp /components controller,desktopstudio /configure_firewall
```

リモートPCアクセス用のVirtual Delivery Agentをインストールする場合は、物理マシン用のオプションと、VDA for Windows Desktop OSをインストールするためのオプションを指定します。仮想マシンやマスターイメージ用のオプションは指定できません。

製品メディアのXenDesktop Setupディレクトリから、XenDesktopVdaSetup.exeコマンドを実行します。次の表は、このコマンドのオプションの一覧です。特に明記されている場合を除き、各オプションは物理マシンおよび仮想マシン、VDA for Windows Desktop OSおよびVDA for Windows Server OSに適用されます。

| オプション                                            | 説明                                                                                                                                                                                                                                                                |
|--------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /hまたは/help                                       | コマンドのヘルプを表示します。                                                                                                                                                                                                                                                   |
| /quietまたは/passive                                | ユーザーインターフェイスを表示せずにインストールを実行します。インストールおよび構成プロセスは、Windowsタスクマネージャーにのみ表示されます。このオプションを指定しない場合、インストールウィザードが表示されます。                                                                                                                                                     |
| /logpath <path>                                  | ログファイルのパスを指定します。既存のフォルダーを指定する必要があります（自動的に作成されません）。デフォルトは、"%TEMP%\CitrixXenDesktopInstaller"                                                                                                                                                                       |
| /noreboot                                        | インストール後の再起動を無効にします Virtual Delivery Agentを使用するには、再起動が必要です。                                                                                                                                                                                                        |
| /remove                                          | /componentsオプションで指定したコンポーネントを削除します。                                                                                                                                                                                                                               |
| /removeall                                       | インストール済みのすべてのコンポーネントを削除します。                                                                                                                                                                                                                                       |
| /reconfig                                        | インストール済みのVirtual Delivery Agent設定をカスタマイズします。/portnumber、/controllers、または/enable_hdx_portsオプションと一緒に使用します。/quietオプションを指定しない場合は、VDAをカスタマイズするためのグラフィカルインターフェイスが開きます。                                                                                                  |
| /portnumber <port>                               | (/reconfigオプションを指定する場合にのみ有効) Virtual Delivery AgentとController間の通信で使用されるポート番号を変更します。変更前のポートは無効になります（ポート80以外は無効になりません）。                                                                                                                                            |
| /components <component>[, <component>]           | インストールまたは削除するコンポーネントをコンマ区切りのリストで指定します。以下の値を指定します。 <ul style="list-style-type: none"> <li>VDA — Virtual Delivery Agentをインストールします</li> <li>PLUGINS — Citrix Receiver for Windows (CitrixReceiver.exe) をインストールします</li> </ul> このオプションを指定しない場合、すべてのコンポーネントがインストールされます。 |
| /installdir <directory>                          | コンポーネントのインストール先として既存の空ディレクトリを指定します。デフォルトは、c:\Program Files\Citrix。                                                                                                                                                                                                |
| /tempdir <directory>                             | インストール時に一時ファイルを作成するディレクトリを指定します（このオプションはグラフィックインターフェイスでは使用できません）。デフォルト値：c:\Windows\Temp                                                                                                                                                                           |
| /site_guid <guid>                                | サイトのActive Directory組織単位（OU）のグローバル一意識別子（GUID）を指定します。Active Directory OUベースのController検出を使用する場合、GUIDにより仮想デスクトップとサイトが関連付けられます（デフォルトの検出方法である自動更新を使用することをお勧めします）。サイトGUIDは、Studioに表示されるサイトプロパティです。/site_guidと/controllersを一緒に指定しないでください。                               |
| /controllers "<controller> [<controller>] [...]" | VDAが通信するControllerの完全修飾ドメイン名（FQDN）を、引用符で囲んだスペース区切りのリストで指定します。/site_guidと/controllersを一緒に指定しないでください。                                                                                                                                                               |
| /xa_server_location <url>                        | アプリケーションを配信するWindowsサーバーのURLです。                                                                                                                                                                                                                                   |
| /enable_remote_assistance                        | Directorで使用するWindowsリモートアシスタンスを有効にします。このオプションを指定すると、/enable_hdx_portsオプションを指定しなくても、WindowsファイアウォールでTCPポート3389が開放されます。                                                                                                                                             |
| /enable_hdx_ports                                | Windowsファイアウォールサービスが実行されている場合に（ファイアウォールが無効になっていても）、Controllerおよび指定する機能（Windowsリモートアシスタンス、リアルタイムオーディオトランスポート、および最適化）に必要なポートが開放されます。Windows以外のファイアウォールを使用している場合は、手作業でファイアウォールを構成する必要があります。                                                                          |
| /optimize                                        | ハイパーバイザー上の仮想マシンにインストールするVDAを最適化します。仮想マシンの最適化には、オフラインファイルの無効化、バックグラウンド最適化（デフラグ処理）の無効化、およびイベントログサイズの縮小などの操作が含まれます。リモートPCアクセスでは、このオプションを指定しないでください。最適化ツールについては、CTX125875を参照してください。                                                                                   |
| /baseimage                                       | （仮想マシン上にVDA for Windows Desktop OSをインストールする場合にのみ有効）マスターイメージでのPersonal vDiskの使用を有効にします。詳しくは、「Personal vDiskの管理」を参照してください。                                                                                                                                          |
| /enable_hdx_3d_pro                               | VDA for HDX 3D Proをインストールします。詳しくは、HDX 3D Proのドキュメントを参照してください。                                                                                                                                                                                                     |
| /enable_real_time_transport                      | オーディオパケットでUDPを使用してパフォーマンスを向上させる機能（リアルタイムオーディオ転送）を有効または無効にします。この機能を有効にすると、オーディオパフォーマンスを向上させることができます。Windowsファイアウォールサービスが検出されたときにUDPポートが開放されるようにするには、/enable_hdx_portsを指定してください。                                                                                    |

|                                                      |                                                                                                                                                                                                                                                       |
|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /masterimage                                         | 仮想マシン上にVDAをインストールする場合にのみ有効) VDAをマスターイメージとしてセットアップします。                                                                                                                                                                                                 |
| /virtualmachine                                      | (仮想マシン上にVDAをインストールする場合にのみ有効) インストーラーによる物理マシンの検出を上書きして、BIOS情報を仮想マシンに渡して物理マシンとして振る舞うようにします。                                                                                                                                                             |
| /nodesktopexperience                                 | (仮想マシン上にVDA for Windows Server OSをインストールする場合にのみ有効) デスクトップエクスペリエンス拡張機能を無効にします。この機能の有効/無効は、Citrixポリシーの [デスクトップエクスペリエンス拡張] 設定でも制御できます。                                                                                                                   |
| /nocitrixwddm                                        | (WDDMドライバーを含まないWindows 7マシン上でのみ有効) Citrix WDDMドライバーのインストールを無効にします。                                                                                                                                                                                    |
| /servervdi                                           | Windowsサーバー上にVDA for Windows Desktop OSをインストールします。VDA for Windows Server OSをインストールする場合は、このオプションを指定しないでください。このオプションを使用する前に、「 <a href="#">サーバーVDI</a> 」を参照してください。<br><br>注：イメージにVDAをインストールしている場合は、/masterimageオプションを追加し、MCSを使ってそのイメージからサーバー仮想マシンを使用します。 |
| /installwithsecurebootenabled                        | セキュアブートが有効なマシンにVDAをインストールします。このオプションを指定しない場合、VDAをインストールするためにセキュアブートを無効にする必要があることを示す警告が表示されます。                                                                                                                                                         |
| /exclude "Personal vDisk","Machine Identity Service" | (物理マシン上のVDA 7.xをアップグレードする場合にのみ有効) Personal vDiskとMachine Identity Serviceをアップグレードから除外します。このオプションについて詳しくは、 <a href="#">CTX140972</a> を参照してください。                                                                                                        |

たとえば、次のコマンドを実行すると、仮想マシン上のデフォルトの場所にVDA for Windows Desktop OSおよびCitrix Receiverがインストールされます。このVDAはマスターイメージとして使用されます。VDAは、まず「mydomain」ドメインの「Contr-Main」サーバー上で動作するControllerに登録され、Personal vDisk、最適化機能、およびWindowsリモートアシスタンスが有効になります。XenDesktop SetupXenDesktopVdaSetup.exe /quiet /components vda,plugins /controllers "Contr-Main.mydomain.local" /enable\_hdx\_ports /optimize /masterimage /baseimage /enable\_remote\_assistanc 次のコマンドを実行すると、リモートPCアクセスで 사용되는社内のPC上のデフォルトの場所にVDA for Windows Desktop OSおよびCitrix Receiverがインストールされます。インストールの完了後にマシンは再起動されませんが、VDAを使用するには再起動が必要です。VDAは、まず「mydomain」ドメインの「Contr-East」サーバー上で動作するControllerに登録され、オーディオパケットにUDPが使用されます。Windowsファイアウォールサービスが検出された場合は、HDXポートが開放されます。  
XenDesktop SetupXenDesktopVdaSetup.exe /quiet /components vda,plugins /controllers "Contr-East.mydomain.local" /enable\_hdx\_ports /enable\_real\_time\_transport /noreboot  
VDAは、XenAppまたはXenDesktop製品のISOからインストールしたり、スタントアロンのVDAインストールパッケージを使用してインストールしたりできます。詳しくは、[スタンドアロンパッケージによるVDAのインストール](#)を参照してください。

VDAをインストールした後で、いくつかの設定をカスタマイズできます。製品メディアの\x64\XenDesktop SetupディレクトリからXenDesktopVdaSetup.exeコマンドを実行します。このとき、以下のオプションを指定します (各オプションについては上の表を参照してください)。

- /reconfigure - VDAをカスタマイズする場合は必須のオプションです。
- /hまたは/help
- /quiet
- /noreboot
- /controllers
- /portnumber <port>
- /enable\_hdx\_ports

注：

- Universal Print Serverの32ビットオペレーティングシステム上での使用はサポートされていません。
- 印刷操作の間にユーザー認証を実行するには、Universal Print ServerがリモートデスクトップサービスVDAと同じドメインに参加する必要があります。
- UPClientコンポーネントは、VDAと一緒にインストールされます。クライアントコンポーネントを手動でインストールする必要はありません。

コマンドラインを使ってUniversal Print Serverをインストールするには

- Windows 2008 R2 SP1、Windows Server 2012、またはWindows Server 2012 R2 R2プリントサーバー上で、LTSRイメージのx64\Universal Print Serverディレクトリから、UpsServer\_x64.msiを実行します。

コマンドラインを使用してUniversal Print Serverをインストールする場合は、コマンドオプション、/ENABLE\_CEIP=1を使用して、[Citrix Customer Experience Improvement Program \(CEIP\)](#) にオプトインすることを推奨します。

次に例を示します。

```
msiexec /i UpsServer.msi ENABLE_CEIP=1
```

参加すると、匿名の統計および使用状況情報が、Citrix製品の品質およびパフォーマンスを向上させる目的で送信されます。


# サイトの作成


Jan 04, 2017

「サイト」とは、Citrix製品の展開環境に名前を付けたものを指します。サイトは、Delivery Controllerなどのコアコンポーネント、VDA、仮想化リソース（使用している場合）、およびそのサイトで作成されたマシンカタログやデリバリーグループ構成されます。サイトの構成は、地理的な場所と必ずしも一致する必要はありません（一致させることもできます）。コンポーネントをインストールしたら、マシンカタログやデリバリーグループを作成する前に、サイトを作成します。

次の表は、Studioのサイトの作成ウィザードを実行する前に完了させておくべきタスク、および注意が必要なタスクについて説明しています。

| ✔                      | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                    |           |         |           |           |                |           |  |         |                                              |                    |          |               |                             |                    |          |                        |                                                      |                    |  |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|-----------|---------|-----------|-----------|----------------|-----------|--|---------|----------------------------------------------|--------------------|----------|---------------|-----------------------------|--------------------|----------|------------------------|------------------------------------------------------|--------------------|--|
|                        | <p>作成するサイトの種類を決定します。</p> <ul style="list-style-type: none"> <li>アプリケーションとデスクトップを配信するサイト：アプリケーションおよびデスクトップの配信サイトを作成する場合は、完全展開サイト（推奨）または空のサイトのいずれかを選択できます（空のサイトには一部の構成のみが含まれ、通常Citrix製品の管理に慣れた管理者がこのオプションを選択します）。</li> <li>リモートPCアクセスサイト：特定のユーザーにオフィスにある自分のコンピューターへのセキュアなリモートアクセスを提供します。リモートPCアクセスのWake on LAN機能を使用する場合は、「<a href="#">Configuration ManagerとリモートPCアクセスのWake on LAN</a>」で説明されているタスクを完了しておく必要があります。</li> </ul> <p>ここでアプリケーションとデスクトップを配信するサイトを作成しても、リモートPCアクセス展開を後で追加できます。また、ここでリモートPCアクセス展開を選択しても、完全展開を後で追加できます。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                    |           |         |           |           |                |           |  |         |                                              |                    |          |               |                             |                    |          |                        |                                                      |                    |  |
|                        | <p>サイトを作成するときに、サイト構成データベースも作成されます。サイトを作成する前に、SQL Serverソフトウェアをインストールしておく必要があります。</p> <p>データベースを作成するには、ローカルの管理者およびドメインユーザーである必要があります。また、SQL Serverの管理権限も必要です。権限がない場合は、データベース管理者に実行させるスクリプトを作成できます。</p> <ul style="list-style-type: none"> <li>権限：データベースをセットアップするときに、以下の権限が必要になります。これらの権限は、個別に設定したり、Active Directoryのグループメンバーシップで構成したりできます。</li> </ul> <table border="1" data-bbox="236 1411 1492 2007"> <thead> <tr> <th>操作</th> <th>目的</th> <th>サーバーロール</th> <th>データベースロール</th> </tr> </thead> <tbody> <tr> <td>データベースの作成</td> <td>空のデータベースを作成します</td> <td>dbcreator</td> <td></td> </tr> <tr> <td>スキーマの作成</td> <td>サービス固有のすべてのスキーマを作成して、サイトに最初のControllerを追加します</td> <td>securityadmin<br/>*</td> <td>db_owner</td> </tr> <tr> <td>Controllerの追加</td> <td>サイトにController（2つ目以降）を追加します</td> <td>securityadmin<br/>*</td> <td>db_owner</td> </tr> <tr> <td>Controller（ミラーサーバー）の追加</td> <td>ミラー化されたデータベースのミラーロールのデータベースサーバーにControllerログインを追加します</td> <td>securityadmin<br/>*</td> <td></td> </tr> </tbody> </table> | 操作                 | 目的        | サーバーロール | データベースロール | データベースの作成 | 空のデータベースを作成します | dbcreator |  | スキーマの作成 | サービス固有のすべてのスキーマを作成して、サイトに最初のControllerを追加します | securityadmin<br>* | db_owner | Controllerの追加 | サイトにController（2つ目以降）を追加します | securityadmin<br>* | db_owner | Controller（ミラーサーバー）の追加 | ミラー化されたデータベースのミラーロールのデータベースサーバーにControllerログインを追加します | securityadmin<br>* |  |
| 操作                     | 目的                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | サーバーロール            | データベースロール |         |           |           |                |           |  |         |                                              |                    |          |               |                             |                    |          |                        |                                                      |                    |  |
| データベースの作成              | 空のデータベースを作成します                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | dbcreator          |           |         |           |           |                |           |  |         |                                              |                    |          |               |                             |                    |          |                        |                                                      |                    |  |
| スキーマの作成                | サービス固有のすべてのスキーマを作成して、サイトに最初のControllerを追加します                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | securityadmin<br>* | db_owner  |         |           |           |                |           |  |         |                                              |                    |          |               |                             |                    |          |                        |                                                      |                    |  |
| Controllerの追加          | サイトにController（2つ目以降）を追加します                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | securityadmin<br>* | db_owner  |         |           |           |                |           |  |         |                                              |                    |          |               |                             |                    |          |                        |                                                      |                    |  |
| Controller（ミラーサーバー）の追加 | ミラー化されたデータベースのミラーロールのデータベースサーバーにControllerログインを追加します                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | securityadmin<br>* |           |         |           |           |                |           |  |         |                                              |                    |          |               |                             |                    |          |                        |                                                      |                    |  |

|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>説明操作の更新 目的の更新およびHotfixを適用します</p> <p>サーバーロール db_owner</p> <p>* securityadminは、技術的にはより限定的なサーバーロールですが、実際にはsysadminサーバーロールと同等のものとして扱われます。</p> <p>Studioを使ってこれらの操作を実行する場合、sysadminサーバーロールの権限が必要です。</p> <p>Studioを使用する管理者にこれらの権限がない場合、SQL Serverユーザーの資格情報を入力する必要があります。</p> <ul style="list-style-type: none"> <li>スクリプト：既存のデータベースサーバーに対する適切なSQL Server権限がない場合は、データベースのセットアップ用のスクリプトおよびミラーリング環境用のスクリプトを生成します。スクリプトを生成したら、適切なSQL Server権限を持つデータベース管理者にそれらのスクリプトを提供してデータベースサーバーおよび必要な場合はミラー化されたデータベース上で実行させます。スクリプトを実行し、データベースの作成に成功したら、サイトの作成を完了できます。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|                                                                                   | <p>ライセンスファイルを後から追加できる30日間無料のトライアルライセンスを使うか、既存のライセンスを使用するか決定します。サイトの作成ウィザード内で、ライセンスファイルをダウンロードしたりライセンスサーバーに追加したりできます。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|                                                                                   | <p>仮想化リソース（ホスト）環境を構成します。</p> <p>XenServerを使用する場合は、以下の点についても考慮してください。</p> <ul style="list-style-type: none"> <li>詳しくは、XenServerのドキュメントを参照してください。</li> <li>VMパワー管理者以上の権限を持つアカウントの資格情報を指定します。</li> <li>XenServerとの通信をHTTPSで保護することをお勧めします。HTTPSを使用するには、XenServer上にインストールされたデフォルトのSSL証明書を信頼される証明機関からの証明書で置き換える必要があります。詳しくは、<a href="#">CTX128656</a>を参照してください。</li> <li>高可用性機能で使用されるハイパーバイザーを選択することもできます（XenServerの高可用性が有効な場合）。プールマスターに障害が生じてもXenServerとの通信が中断されないように、プール内のすべてのサーバーを選択することをお勧めします。追加したホストの"Edit High Availability"から選択できます。</li> <li>XenServerでvGPUがサポートされる場合は、GPUの種類およびグループ、またはGPUパスルーを選択することもできます。選択した項目で専用のGPUリソースが使用可能かどうか画面に表示されます。</li> </ul> <p>VMwareを使用する場合は、製品ドキュメントおよび「<a href="#">仮想化環境の準備：VMware</a>」を参照してください。</p> <p>Hyper-Vを使用する場合は、製品ドキュメントおよび「<a href="#">仮想化環境の準備：Microsoft System Center Virtual Machine Manager</a>」を参照してください。</p> <p>仮想化リソース上の仮想マシンを作成するツールとして、Machine Creation Services（MCS）を使用するかそのほかのツールを使用するかを決定します。</p> <p>共有ストレージを使用するかローカルストレージを使用するかを決定します。共有ストレージはネットワーク経由で使用します。共有ストレージを使用する場合は、IntelliCache機能を有効にしてストレージデバイス上の負荷を減らすことができます。詳しくは、「<a href="#">XenServer接続でのIntelliCacheの使用</a>」を参照してください。</p> <p>Personal vDiskを使用するかどうか、使用する場合は共有ストレージに格納するかローカルストレージに格納するかを決定します。Personal vDiskを仮想マシンと同じストレージに格納したり、別のストレージに格納したりでき</p> |

|                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p><b>説明。</b></p> <p>クラウド環境に製品コンポーネントをインストールした場合は、最初の接続の構成時にAPIキーと秘密キーが必要になります。AWSまたはCloudPlatformでこれらの値を含んでいるキーファイルをエクスポートして、サイトの作成ウィザードでインポートできます。</p> <p>クラウド用のサイトを作成する場合は、AWSで構成したリージョン、アベイラビリティゾーン、VPC名、サブネットアドレス、ドメイン名、セキュリティグループ名、および資格情報が必要になります。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|                                                                                   | <p>App-V公開を使用するかどうかを決定し、必要に応じてそれらのリソースを構成します。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|                                                                                   | <p>ヒント：</p> <ul style="list-style-type: none"> <li>● リモートPCアクセスサイトを作成する場合： <ul style="list-style-type: none"> <li>● 「リモートPCアクセスマシン」という名前のマシンカタログと、「リモートPCアクセスデスクトップ」という名前のデリバリーグループが自動的に作成されます。</li> <li>● ユーザーまたはユーザーグループを指定する必要があります。自動的にすべてのユーザーを追加するデフォルトの機能はありません。</li> <li>● Wake on LAN機能（電源管理）を有効にしたり、Microsoft System Center Configuration Manager (ConfigMgr) アドレス、資格情報、および接続名を指定したりできます。</li> </ul> </li> <li>● サイトを作成する管理者には、そのサイトのすべての管理タスクの実行権限が設定されます。詳しくは <a href="#">管理権限の委任</a> を参照してください。</li> <li>● 空のデータベースを作成すると、デフォルトの属性が設定されます。ただし、以下の属性を除きます。 <ul style="list-style-type: none"> <li>● 照合順序には、Latin1_General_100_CI_AS_KSが設定されます（「Latin1_General」の部分は国により異なります。例：Japanese_100_CI_AS_KS）。データベースの作成時にこの照合順序設定を指定しないと、そのデータベースでのサービスのスキーマ作成に失敗し、「：スキーマには大文字と小文字を区別しないデータベースが必要です」というエラーメッセージが表示されます（データベースを手動で作成する場合は、大文字と小文字の区別、アクセントの区別、およびかなの区別が有効な任意の照合順序を使用できます。照合順序の名前には、通常最後に_CI_AS_KSが付きます）。</li> <li>● 回復モードはSimpleに設定されます。ミラー化されたデータベースとして使用する場合は、回復モードをFullに変更してください。</li> </ul> </li> <li>● サイト構成データベースには、Configuration Logging Serviceで記録される構成の変更データと、Monitoring ServiceやCitrix Directorで使用される傾向データおよびパフォーマンスデータが格納されます。これらの機能のデータを8日以上保持する場合は、サイトを作成した後で、構成ログデータベースと監視データベース（セカンダリデータベースと呼ばれます）を別の場所に配置することをお勧めします。</li> <li>● 監視データベースまたは監視データベースを含むサイト構成データベースに名前を付ける場合、名前にスペースを含めるとデータベースへのアクセス時にエラーが発生します。詳しくは、<a href="#">CTX200325</a>を参照してください。</li> <li>● サイトの作成ウィザードの最後に、Citrixのカスタマーエクスペリエンス向上プログラムに参加するかどうかを確認するメッセージが表示されます。このプログラムに参加すると、統計情報や使用状況が匿名でCitrixに送信されます。詳しくは、「<a href="#">Citrixカスタマーエクスペリエンス向上プログラムについて</a>」を参照してください。</li> </ul> |

Studioが起動していない場合は起動します。中央ペインでサイトの作成を選択した後、以下の項目を指定します。

- サイトの種類およびサイト名。
- データベース情報。Controllerのインストール時にデフォルトのSQL Server Expressデータベースのインストールを選択し

た場合は、一部の情報が自動的に指定されます。別のサーバーにインストール済みのデータベースサーバーを使用する場合は、データベースサーバーの場所と名前を入力します。

| データベースの種類         | 入力内容                            | データベースの構成                                                 |
|-------------------|---------------------------------|-----------------------------------------------------------|
| スタンドアロン<br>またはミラー | servername                      | デフォルトのインスタンスが使用され、SQL Serverでデフォルトのポートが使用されている場合。         |
|                   | <servername>\<br><INSTANCENAME> | 名前付きのインスタンスが使用され、SQL Serverでデフォルトのポートが使用されている場合。          |
|                   | <servername>,<port-<br>number>  | デフォルトのインスタンスが使用され、SQL Serverでカスタムポートが使用されている場合（コンマが必要です）。 |
| そのほか              |                                 | データベースがクラスター化されている場合。                                     |
|                   |                                 | AlwaysOnデータベースが構成されている場合。                                 |

[次へ] をクリックします。新規のデータベースを使用する場合は、データベースに接続できないこと示すメッセージが表示されます。ここでデータベースを作成するためのオプションを選択します。データベースを編集する権限がない場合は、[データベーススクリプトを作成] を使用します。このスクリプトは、サイトを作成する前に実行する必要があります。

- ライセンスサーバーのアドレス。「[:]」形式で指定します。ここで、<name>は完全修飾ドメイン名（FQDN）、NetBIOS、またはIPアドレスです（FQDNを推奨）。ポート番号（）を入力しない場合は、デフォルトの27000が使用されます。ライセンスサーバーに接続されるまでは、先に進めません。
- （リモートPCアクセスサイトのみ）電源管理情報。ConfigMgrの接続情報など。
- 仮想化リソースの接続情報とストレージの情報。仮想化リソースを使用しない場合、または専用ブレードPC上でホストされるユーザーデスクトップをStudioで管理する場合は、接続の種類として[なし]を選択します。
- App-V管理サーバーおよびApp-V公開サーバーの情報。
- （リモートPCアクセスサイトのみ）ユーザーおよびマシンアカウントの情報。
  - ユーザー情報。[ユーザーの追加] をクリックします。ユーザーおよびユーザーグループを選択し、[ユーザーの追加] をクリックします。
  - マシンアカウント情報。[マシンアカウントの追加] をクリックします。マシンアカウントを選択し、[マシンアカウントの追加] をクリックします。[組織単位の追加] をクリックします。ドメインおよび組織単位を選択して、サブフォルダー内の項目を含めるかどうかを指定します。[組織単位の追加] をクリックします。

テスト結果は、HTML形式のレポートで確認できます。テストを実行するには、次の手順に従います。

1. Studioで、左ペイン上部にある[Citrix Studio (サイト名)] ノードをクリックします。
2. 中央のペインで、[サイトのテスト] をクリックします。

# スクリプトを使用したVirtual Delivery Agentのインストールと削除

Jan 04, 2017

この製品のインストールメディアには、Active Directory環境のマシンのグループでVirtual Delivery Agent (VDA) をインストール、アップグレード、または削除するサンプルスクリプトが収録されています。また、このスクリプトを個々のマシンに適用して、それらを使ってMachine Creation ServicesおよびProvisioning Servicesのマスターイメージを管理することもできます。

以下のアクセス権限が必要です。

- スクリプトを実行するには、VDAインストールコマンドがあるネットワーク共有に対するすべてのユーザーの読み取りアクセスが必要です。インストールコマンドはフルバージョンの製品ISOのXenDesktopVdaSetup.exe、またはスタンドアロンインストーラーのVDAWorkstationSetup.exeまたはVDA ServerSetup.exeです。
- ログの詳細は各ローカルマシンに保存されます。また、レビューおよび分析のために結果ログをネットワーク上に保存する場合は、そのネットワーク共有に対するすべてのユーザーの読み取りおよび書き込みアクセスが必要です。

スクリプトの実行結果をチェックするには、ネットワーク共有のログを調べます。このログには、スクリプトログ、インストーラーログ、およびMSIインストールログが含まれます。各インストールまたは削除に関するログは、日時を示すフォルダー内に保存されます。フォルダー名には、操作が成功した場合にはプレフィックスとしてPASSが付き、そのほかの場合はFAILが付きます。失敗したインストールまたは削除処理をすばやく検索できるように、ターゲットマシンのローカルドライブではなく、ネットワーク共有を使用します。

重要：インストールを始める前に、「[インストールの準備](#)」を読んで、必要なタスクを完了しておいてください。

1. インストールメディアの\Support\AdDeploy\からサンプルスクリプトのInstallVDA.batを開きます。スクリプトをカスタマイズする前に、元のスクリプトをバックアップしておくことをお勧めします。
2. スクリプトを編集します。
  - インストールするVDAのバージョンを指定するため、SET DESIREDVERSIONと入力します。Version 7の場合はバージョンを「7.0」と指定できます。完全な値はインストールメディアのProductVersion.txtファイルに記述されています（7.0.0.3018など）。ただし、完全に一致させる必要はありません。
  - 実行するインストーラーのネットワーク共有を指定します。レイアウトのルート（ツリーの最上位）を指定します。スクリプトにより、適切なバージョンのインストーラー（32ビットまたは64ビット）が自動的に実行されます。たとえば、SET DEPLOYSHARE=\\fileserver1\share1と入力します。
  - オプションとして、ログを保存するためのネットワーク共有を指定します。たとえば、SET LOGSHARE=\\fileserver1\log1と入力します。
  - 「[コマンドラインを使ったインストール](#)」の説明に従って、VDAの構成オプションを指定します。/quietおよび/norebootオプションはデフォルトでスクリプトに含まれ、必須です。SET COMMANDLINEOPTIONS=/QUIET /NOREBOOTと入力します。
3. グループポリシースタートアップスクリプトを使って、マシンが存在するActive Directoryの組織単位にスクリプトを割り当てます。VDAをインストールするマシン以外のもがこの組織単位に属していないことを確認してください。組織単位のマシンの再起動時にスクリプトが実行され、サポートされるオペレーティングシステムの各マシン上にVDAがインストールされます。

1. インストールメディアの\Support\AdDeploy\からサンプルスクリプトのUninstallVDA.batを開きます。スクリプトをカスタ



マイズする前に、元のスクリプトをバックアップしておくことをお勧めします。

2. スクリプトを編集します。
  - 削除するインストールするVDAのバージョンを指定するため、SET CHECK\_VDA\_VERSIONと入力します。Version 7の場合はバージョンを「7.0」と指定できます。完全な値はインストールメディアのProductVersion.txtファイルに記述されています(7.0.0.3018など)。ただし、完全に一致させる必要はありません。
  - オプションとして、ログを保存するためのネットワーク共有を指定します。
3. グループポリシースタートアップスクリプトを使って、マシンが存在するActive Directoryの組織単位にスクリプトを割り当てます。VDAを削除するマシン以外のものがこの組織単位に属していないことを確認してください。組織単位内のマシンの再起動時にスクリプトが実行され、各マシンからVDAが削除されます。

スクリプトにより、スクリプトの進捗を示す内部ログファイルが生成されます。スクリプトは、マシンへの展開の開始後すぐにKickoff\_VDA\_Startup\_Scriptログをネットワーク共有にコピーします。これにより、処理全体が実行中であることを確認できます。このログがネットワーク共有にコピーされない場合は、ターゲットマシンのローカルドライブを調べることでトラブルシューティングを実行できます。初期トラブルシューティングのため、各マシン上の%temp%フォルダーに以下の2つのデバッグログファイルが生成されます。

- Kickoff\_VDA\_Startup\_Script\_log
- VDA\_Install\_ProcessLog\_log

これらのログの内容から、次の点を確認します。

- スクリプトが正しく実行されたかどうか。
- ターゲットのオペレーティングシステムが正しく検出されているかどうか。
- DEPLOYSHARE共有でROOT (AutoSelect.exeファイルを含んでいるフォルダー) が正しく構成されているかどうか。
- DEPLOYSHAREおよびLOGで指定した両方のネットワーク共有にアクセスできるかどうか。

# スタントアロンパッケージによるVDAのインストール

Jan 04, 2017

Virtual Delivery Agent (VDA) は、XenAppまたはXenDesktop製品のISOからインストールしたり、スタントアロンのVDAインストールパッケージを使用してインストールしたりできます。スタントアロンパッケージはサイズが小さいため、電子ソフトウェア配信 (ESD) ツールを使用してVDAをローカルにステージングまたはコピーしたり、物理マシンやリモートオフィスに展開したりする場合に便利です。

VDAのスタントアロンパッケージは主にコマンドラインでのサイレント (無人) インストールで使用し、XenAppまたはXenDesktop製品のISOに付属のXenDesktopVdaSetup.exeコマンドと同じコマンドラインパラメーターを指定できます。このパッケージには、製品ISOのVDAインストーラーに似たグラフィカルインターフェイスも用意されています。

VDAの自己展開型スタントアロンパッケージには、サーバーOSマシン用とワークステーション (デスクトップ) OSマシン用の2つがあります。

VDAでサポートされるオペレーティングシステムおよびそのほかのインストール要件については、[XenApp 7.6](#)、[XenDesktop 7.6](#)、[XenApp 7.6 FP1](#)、および[XenDesktop 7.6 FP1のシステム要件](#)を参照してください。VDAのインストール時に指定する情報については「[インストールの準備](#)」を参照してください。

**XenAppおよびXenDesktop 7.6 FP3。**必要なサードパーティ製ソフトウェアがインストールされていない場合は、VDAパッケージにより自動的にインストールされます。これにはVisual C++ 2008、2010、2013のRuntime (32ビットと64ビット)、および.NET Framework 4.5.1などがあります。

この.NET要件は、VDAをインストールするデスクトップにWindows 10 (.NET 4.6を含む) がインストールされている場合は、インストーラーによってスキップされます。

**XenAppおよびXenDesktop 7.6、7.6 FP1、7.6 FP2。**必要なサードパーティ製ソフトウェアがインストールされていない場合は、VDAパッケージにより自動的にインストールされます。これにはMicrosoft Visual C++ 2005、2008、2010のRuntime (32ビットと64ビット)、および.NET Framework 4.5.1などがあります。

サーバーOSマシンでのインストールでは、VDAのインストール前にリモートデスクトップサービス (RDS) の役割サービスがインストールされて有効になります。これらの必須条件を手作業でインストールしてからVDAをインストールすることもできます。

例外：Windows Server 2008 R2およびWindows 7では、.NET 3.5.1またはそれ以降がインストールされていることを確認してからVDAをインストールしてください。

## 再起動について

- VDAのインストールプロセスの最後にマシンを再起動する必要があります。
- インストールシーケンスで再起動する回数を最小化するには、.NET Framework 4.5.1または4.5.2をインストールしてから、VDAインストールを開始するようにします。WindowsサーバーOSマシンでは、RDSの役割サービスをインストールして有効にしてからVDAをインストールしてください。通常、ほかの必須条件のインストールでは再起動が不要なため、インストーラーでこれらのソフトウェアを自動的にインストールできます。
- VDAをインストールする前に必須条件をインストールせず、コマンドラインで/norebootオプションを指定してインストールする場合は、必要に応じてマシンを再起動してください。たとえば、必須条件を自動的にインストールする場合は、RDのインストール後にインストーラーがいったん停止して再起動を確認するメッセージが表示されます。VDAのインストールを続行するには、マシンを再起動した後でインストールコマンドを再実行してください。

インストーラーのグラフィカルインターフェイスを使用したりパッケージを実行するコマンドラインインターフェイスを使用したりする場合、パッケージに含まれているファイルはTempフォルダーに抽出されます。Tempフォルダーに抽出する場合は、フル製品ISOを使用する場合よりも多くの空き領域が必要です。Tempフォルダーに抽出されたファイルは自動的に削除されません。インストールが完了したら、これらのファイルをC:\Windows\Temp\Ctx-\*（ここで「\*」はランダムなグローバル一意識別子）から手作業で削除することができます。

また、サードパーティ製のキャビネットアーカイブ抽出ツール（7-Zipなど）を使用して、EXEファイルから任意のフォルダーにファイルを抽出して、それらに含まれているXenDesktopVdaSetup.exeコマンドを実行することもできます。7.6 FP3では、絶対パスと/extractコマンドを使用できます。詳しくは、後述の使い方についての説明を参照してください。

Microsoft System Center Configuration Managerを使用する環境では、VDAのインストールに成功しても終了コード3により失敗したというメッセージが表示されることがあります。この不正なメッセージが表示されなくなるようにするには、インストールコマンドをCMDスクリプト内に記述するか、Configuration Managerパッケージの成功コードを変更してください。詳しくは、[このサポートフォーラム](#)を参照してください。

### XenAppおよびXenDesktop 7.6 FP3のCitrix Display Ony Driver

Citrix Display Only Driver (DOD) は、Windows 10上のXenDesktop Standard VDAにインストールされてサポートされる唯一のディスプレイドライバーです。

GPUまたはvGPUがある場合でも、Citrix DODにはGPUアシストがありません。すべてのレンダリングは、CPUを使用するソフトウェアのMS Basic Rendererで実行されます。Citrix DODはDesktop Composition Redirection (DCR) をサポートしていません。Citrix DODはXenAppにはインストールされず、サポートされません。

**重要：**インストールは、管理者権限または[管理者として実行]で実行する必要があります。

1. VDAのインストール先のマシンに適切なパッケージをダウンロードします。ダウンロードサイトにアクセスするには、Citrixアカウントの資格情報が必要です。

| VDAのインストール先            | ダウンロードするパッケージ            |
|------------------------|--------------------------|
| サーバーOSマシン              | VDA ServerSetup.exe      |
| ワークステーション（デスクトップ）OSマシン | VDA WorkstationSetup.exe |

単一ユーザー、単一サーバーOSマシン展開（たとえばWindows Server 2012マシンをWebサイト開発者などの単一ユーザーに配信する場合など）では、VDAWorkstationSetup.exeパッケージを使用します。詳しくは、「[サーバーVDI](#)」を参照してください。

2. グラフィカルインターフェイスまたはコマンドラインインターフェイスを使用して、VDAをインストールします。

**注意：**インストールは、管理者権限または[管理者として実行]で実行する必要があります。

**グラフィカルインターフェイスの使用：**

ユーザーアカウントコントロール (UAC) を無効にし、次にダウンロードしたパッケージを右クリックして、[管理者として実行]を選択します。インストールウィザードの指示に従って操作します。ウィザードの最後でマシンを再起動すると、VDAが使用可能になります。このウィザードでは、製品ISOのVDAインストーラーと同じ画面が開きます。

**コマンドラインインターフェイスの使用：**

以下のいずれかの方法を使用します。

- パッケージの内容を抽出してXenDesktopVdaSetup.exeを実行する。

XenAppおよびXenDesktop 7.6 FP3。インストール前にファイルを抽出するには、絶対パスと/extractを使います。  
例：

```
.\VDAWorkstationSetup.exe /extract %temp%\CitrixVDAInstallMedia
```

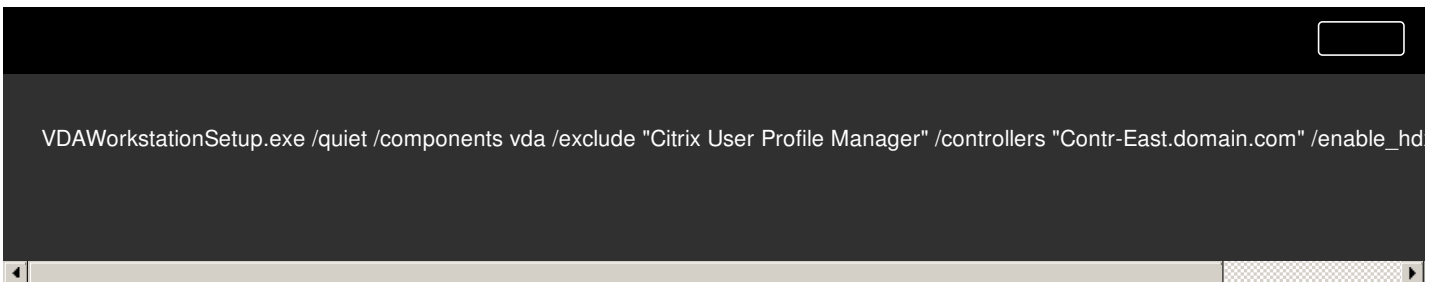
次に、新しいコマンドプロンプトを開いて、抽出先フォルダーからXenDesktopVdaSetup.exeを実行します。詳しくは、「[コマンドラインを使ったインストール](#)」およびパラメーター情報については[CTX140972](#)を参照してください。

XenAppおよびXenDesktop 7.6、7.6 FP1、7.6 FP2。EXEファイルからキャビネットアーカイブを抽出できるサードパーティ製のツール（7-Zipなど）を使用して、インストール前にファイルを抽出します。次に、新しいコマンドプロンプトを開いて、抽出先フォルダーからXenDesktopVdaSetup.exeを実行します。詳しくは、「[コマンドラインを使ったインストール](#)」およびパラメーター情報については[CTX140972](#)を参照してください。

- パッケージを実行する。

ダウンロードしたパッケージを、XenDesktopVdaSetup.exeコマンドと同じ方法で実行します（同じパラメーターを使用しますが、コマンド名のみ異なります）。詳しくは、「[コマンドラインを使ったインストール](#)」およびパラメーター情報については[CTX140972](#)を参照してください。

たとえば一般的なリモートPCアクセス用のインストールでは、オフィスにある物理PC上にVDAのみをインストールして、Citrix ReceiverやCitrix Profile Managerはインストールしません。以下の例でインストール後にマシンは再起動されませんが、VDAを使用するには再起動が必要です。VDAは「Contr-East」という名前のControllerに登録されます。インストール時にWindowsファイアウォールサービスが検出された場合は、適切なポートが開放されます。



```
VDAWorkstationSetup.exe /quiet /components vda /exclude "Citrix User Profile Manager" /controllers "Contr-East.domain.com" /enable_hd
```

## 注意

インストールからCitrix Profile Managementを除くと（/exclude "Citrix User Profile Manager"オプションを使用）、Citrix Directorを使ったVDAの監視やトラブルシューティングに影響があります。[ユーザーの詳細] ページの [個人設定] パネル、および [EndPoint] ページの [ログオン処理時間] パネルに不具合が発生します。[ダッシュボード] ページと [傾向] ページでは、Profile managementがインストールされているマシンについてのデータしか [平均ログオン処理時間] パネルに表示されません。

サードパーティのユーザープロファイル管理ソリューションを使用している場合でも、Citrix Directorで監視およびトラブルシューティングのエラーを回避するために、Citrix Profile Managementサービスをインストールして実行することをお勧めします（Citrix Profile Managementサービスの有効化は、必須ではありません）。

# マシンカタログ

Jan 04, 2017

物理マシンまたは仮想マシンのグループは、「マシンカタログ」と呼ばれる単一のエンティティとして管理されます。通常、ホスト上にマスターイメージまたはテンプレートを作成し、マシンカタログでそれらを基にしてCitrixツール（Machine Creation ServicesやProvisioning Servicesなど）を使用して仮想マシンを作成します。カタログには物理マシンを含めることもできます。

マシンカタログを作成すると、テストが自動的に実行されて正しく構成されているかどうかを検証されます。テストが完了したら、テストレポートを表示できます。また、Studioのナビゲーションペインの [Citrix Studio<サイト名>] では、必要に応じてこれらのテストを実行することもできます。

テストが完了したら、[デリバリーグループを作成します](#)。

# マシンカタログの作成

Jan 04, 2017

仮想マシンの作成ツールとしてMachine Creation ServicesまたはProvisioning Servicesを使用する場合は、ハイパーバイザーホストでマスターイメージまたはテンプレートを準備します。その後で、マシンカタログを作成します。

作成するマシンの数に応じて十分なプロセッサ、メモリ、ストレージがハイパーバイザーホスト上にあることを確認してください。

マスターイメージには、オペレーティングシステム、仮想化しないアプリケーション、VDA、およびその他のソフトウェアをインストールしておきます。マシンカタログの仮想マシンは、そのカタログの作成時に指定した既存のマスターイメージに基づいて作成されます。

ヒント：

- マスターイメージは「クローンイメージ」、「ゴールデンイメージ」、または「基本イメージ」とも呼ばれます。
- クラウド展開では、マスターイメージではなくテンプレートを使用します。テンプレートについては、ガイダンスを参照してください。
  - Amazon Web Services (AWS) — 「[Deploy XenApp and XenDesktop 7.5 and 7.6 with Amazon VPC](#)」を参照してください。
  - Citrix CloudPlatform — 「[XenApp and XenDesktop concepts and deployment on CloudPlatform](#)」を参照してください。

Provisioning Servicesを使用する場合は、マスターイメージまたは物理コンピューターをマスターターゲットデバイスとして使用できます。

- リモートPCアクセスのマシンカタログでは、マスターイメージを使用しません。
- Machine Creation Servicesを使用する場合のMicrosoft KMSのアクティブ化については、以下の点に注意してください。
  - VDA 7.xをXenServer 6.1、XenServer 6.2、vSphere、またはMicrosoft System Center Virtual Machine Managerホストで使用している場合、Microsoft WindowsやMicrosoft Officeのライセンスを手動でリセットする必要はありません。
  - VDA 5.xをXenServer 6.0.2ホストで使用している場合は、[CTX128580](#)を参照してください。

**重要：**Provisioning ServicesまたはMachine Creation Servicesを使用する場合は、マスターイメージ上でMicrosoft System Preparation Utility (Sysprep) を実行しないでください。

1. ハイパーバイザーの管理ツールを使用して、新しいマスターイメージを作成してから、オペレーティングシステムと、すべてのサービスパックおよび更新プログラムをインストールします。  
この時点では、vCPUの数とメモリ量の設定は重要ではありません。これらの値はマシンカタログを作成するときに変更できます。ただし、デスクトップやアプリケーションの実行に必要なハードディスク容量をこの時点で指定しておく必要があります。この値は後から変更できません。
2. ハードディスクは、デバイスの場所「0」で接続されている必要があります。多くの標準マスターイメージテンプレートでは、デフォルトでこの場所にハードディスクが構成されますが、カスタムテンプレートを使用する場合は注意してください。
3. マスターイメージに以下のソフトウェアをインストールして構成します。
  - ハイパーバイザー用の統合ツール (XenServer Tools、Hyper-V統合サービス、VMware Toolsなど)。この手順を省略すると、アプリケーションやデスクトップが正しく動作しなくなる場合があります。
  - VDA for Windows Server OSまたはVDA for Windows Desktop OS (最新の機能を利用できるように最新バージョンをイ

インストールすることをお勧めします)。インストール時に最適化オプションを有効にすると、特定のWindows機能が再構成されてパフォーマンスが向上します。

- アンチウイルスプログラムや電子ソフトウェア配信エージェントなどのサードパーティツール（必要に応じて）。ユーザーやマシンの種類に適した設定でWindows Updateなどのサービスを構成します。
  - 仮想化せずにユーザーに提供するサードパーティのアプリケーション。ただし、可能な場合はアプリケーションを仮想化することをお勧めします。アプリケーションを仮想化して配信することで、ユーザーのデスクトップにアプリケーションを追加したり再構成したりするたびにマスターイメージを更新する必要がなくなり、デスクトップの管理コストが削減されます。また、各デスクトップにインストールするアプリケーションが少なくなるため、マスターイメージのハードディスクのサイズを減らしてストレージコストを節約できます。
  - App-Vアプリケーションを公開する場合は、推奨設定のApp-Vクライアント。
  - Machine Creation Servicesで作成したマシンカタログで、ローカライズされたMicrosoft Windowsを配信する場合は、マスターイメージに言語パックをインストールして言語オプション（システムロケールや表示言語など）を設定しておく必要があります。これにより、プロビジョニング時にスナップショットが作成されると、その言語パックおよび言語オプションが仮想マシンで使用されます。
4. Provisioning Servicesを使用する場合は、マスターターゲットデバイスをドメインに追加する前に、マスターターゲットデバイスから作成したvDiskのVHDファイルを作成します。
  5. マスターイメージを、そのマスターイメージで提供するデスクトップやアプリケーションのドメインに追加して、仮想マシンを作成するハイパーバイザーホスト上でこのマスターイメージを使用できることを確認します。
  6. マスターイメージのスナップショットを作成して、わかりやすい名前を付けておくことをお勧めします。マシンカタログの作成時にスナップショットの代わりにマスターイメージを指定すると、Studioによりスナップショットが作成されますが、そのスナップショットにわかりやすい名前を付けることはできません。

**XenServerでのGPU対応マシン用マスターイメージの準備** — ホストインフラストラクチャとしてXenServerを使用する場合は、GPU対応マシンに専用のマスターイメージが必要です。これらの仮想マシンでは、GPUをサポートするビデオカードドライバをインストールして、グラフィックソフトウェアでGPUを使用できるように構成しておく必要があります。

1. XenCenterを使用して、標準的なVGA、ネットワーク、およびvCPUを指定して仮想マシンを作成します。
2. 作成した仮想マシンの構成を変更して、GPU機能（パススルーまたは仮想GPU）を有効にします。
3. 仮想マシンに適切なオペレーティングシステムをインストールして、RDPを有効にします。
4. XenServer ToolsとNVIDIAドライバをインストールします。
5. パフォーマンスを最適化するため、Virtual Network Computing (VNC) Admin Consoleをオフにして、仮想マシンを再起動します。
6. RDPの使用を確認するメッセージが表示されます。RDPを使用してVDAをインストールし、仮想マシンを再起動します。
7. 必要に応じて、仮想マシンのスナップショットを作成します。このスナップショットは、ほかのGPUマスターイメージのテンプレートとして使用できます。
8. RDPを使用して、GPUを使用するアプリケーションを仮想マシンにインストールします。

マシンカタログの作成ウィザードを開始する前に、以下の手順を確認して、選択する項目および指定する情報について理解しておいてください。環境やこのウィザードで選択する項目によっては、オプションが表示されなかったりタイトルが異なったりする場合があります。

Studioで以下の操作を行います。

- サイトを作成した後でマシンカタログを作成していない場合は、マシンカタログを作成するための説明が表示されます。
- 既存のマシンカタログがあり、別のマシンカタログを作成する場合は、Studioのナビゲーションペインで[マシンカタログ]を選択し、[操作]ペインの[マシンカタログの作成]を選択します。

ウィザードの指示に従って、以下の操作を行います。

- オペレーティングシステム

各カタログでは、以下のいずれかの種類のマシンを追加します。

- WindowsサーバーOSマシン：WindowsサーバーOSマシンのカタログでは、複数のユーザーで共有可能なデスクトップやアプリケーションが提供されます。
- WindowsデスクトップOSマシン：WindowsデスクトップOSマシンのカタログでは、個々のユーザーに割り当てられるデスクトップやアプリケーションが提供されます。
- リモートPCアクセス：リモートPCアクセスのカタログでは、オフィスにあるユーザーの物理デスクトップマシンへのリモートアクセスが提供されます。リモートPCアクセスでは、セキュリティを保護するためのVPNが不要です。

Amazon Web Services (AWS) ではサーバーOSマシンカタログとサーバーVDI (「[サーバーVDI](#)」を参照) のみがサポートされます。デスクトップOSカタログやリモートPCアクセスカタログはサポートされません。

#### ● マシン管理

Studioを使用してカタログ内のマシンの電源を管理するかどうかを指定します。

- Studioで電源管理したりクラウド環境でプロビジョニング (仮想マシンやブレードPCなど) したりするマシン。このオプションは、ハイパーバイザーやクラウド環境接続が構成済みの場合にのみ使用可能です。通常、これらの接続はサイトを作成するときに構成します。接続が構成済みでない場合は、マシンカタログの作成後に新しい接続を作成してから、そのマシンカタログを編集できます。
- Studioで電源管理しないマシン (物理マシンなど)。

マシンの展開に使用するツールを指定します。

- Machine Creation Services (MCS) — マスターイメージまたはテンプレートを使用して仮想マシンを作成および管理します。
  - MCSは物理マシンでは使用できません。
  - クラウド環境内のマシンカタログではMCSが使用されます。
- Provisioning Services — 複数のターゲットデバイスを単一のデバイスコレクションとして管理します。マスターターゲットデバイスからイメージ作成されたProvisioning Services vDiskを使用して、デスクトップとアプリケーションを配信します。
- ほかのサービスまたはテクノロジー — 上記以外のツールでデータセンター内の既存のマシンを管理します。この場合、Microsoft System Center Configuration Managerまたはほかのサードパーティアプリケーションを使用してカタログ内のマシン構成を一貫させることをお勧めします。

#### ● デスクトップエクスペリエンス

デスクトップを配信するデスクトップOSマシン用のマシンカタログでは、以下のオプションを指定します。

- ユーザーがログオンするたびに新しいデスクトップに接続する (ランダム) のか、またはログオンするたびに同じデスクトップに接続する (静的) のかを指定します。
- 同じデスクトップに接続する場合は、ユーザーによるデスクトップへの変更を保持するかどうかを指定します。ユーザーによる変更内容は、専用のPersonal vDiskまたはユーザーのローカル仮想マシンディスクに保存したり、破棄したりできます (Personal vDiskに保存する場合は、このウィザードの後のページでドライブ文字とサイズを指定します)。

#### ● マスターイメージまたはマシンテンプレート

既存のマスターイメージ (非クラウド環境) またはマシンテンプレート (クラウド環境) を選択します。Provisioning ServicesまたはMachine Creation Servicesを使用する場合は、マスターイメージ上でMicrosoft System Preparation Utility (Sysprep) を実行しないでください。

#### ● セキュリティ

(クラウド環境) 仮想マシンの1つまたは複数のセキュリティグループを選択します。これらはアベイラビリティゾーンでセキュリティグループがサポートされる場合にのみ表示されます。マシンで使用するハードウェアとして、共有のハードウェアまたはアカウント専用のハードウェアを選択します。

#### ● 仮想マシン、デバイスコレクションまたは仮想マシンとユーザー

作成する仮想マシンの数を指定します。各マシンで使用する仮想CPUの数とメモリ量 (MB単位) を選択できます。各仮想マシンには32GBのハードディスクがあります。この値はマスターイメージで設定され、マシンカタログでは変更できません。



ん。

ユーザーによるデスクトップへの変更を専用のPersonal vDiskに保存することを指定した場合は、そのサイズ（GB単位）とドライブ文字を指定します。

複数のネットワークインターフェイスカード（NIC）を使用する場合は、各NICに仮想ネットワークを関連付けます。たとえば、特定のセキュアネットワークへのアクセスに1枚のNICを割り当てて、より一般的なネットワークへのアクセスに別のNICを割り当てることができます。また、このウィザードでNICを追加または削除することもできます。

- マシンアカウント

（リモートPCアクセスカタログ）ユーザーまたはユーザーグループに対応するActive Directoryマシンアカウントまたは組織単位（OU）を指定して追加します。

構成済みの電源管理接続を選択するか、電源管理を使用しないことを選択します。電源管理に必要な接続が構成済みでない場合は、マシンカタログの作成後に新しい接続を作成してから、そのマシンカタログを編集して電源管理設定を更新できます。

- コンピューターアカウント

カタログ内の各マシンには、対応するActive Directoryコンピューターアカウントを割り当てる必要があります。新しいアカウントを作成するか既存のものを選択して、アカウントの場所を指定します。

既存のアカウントを使用する場合は、作成するマシン用に十分な数のコンピューターアカウントが使用可能かどうかを確認してください。

Active Directoryを参照して既存のアカウントを指定したり、CSVファイル形式のアカウント名の一覧をインポートしたりできます。インポートするファイルでは、次の形式を使用する必要があります。

[ADComputerAccount] ADcomputeraccountname.domain ...

物理マシンまたは既存のマシン用のカタログでは、既存のアカウントを選択またはインポートして、各マシンをActive Directoryコンピューターアカウントおよびユーザーアカウントに割り当てます。

Provisioning Servicesで作成されたマシンでは、ターゲットデバイスのコンピューターアカウントは異なる方法で管理されます。詳しくは、Provisioning Servicesのドキュメントを参照してください。

また、作成されるマシンのアカウント名前付けスキームを指定します。スキームの番号記号（#）は、アカウント名に追加される連番また文字を表します。

- 名前と説明

作成ウィザードの最後のページで、マシンカタログの名前と説明を指定します。これらの情報はStudioに表示されます。

# マシンカタログの管理

Jan 04, 2017

ランダムに割り当てられるマシンカタログの場合、Windows Update、アンチウイルスソフトウェアの更新、または構成の変更など、全体的な更新をマスターイメージに適用して、ユーザーのデスクトップを保守します。その後でマシンカタログを編集して、更新したマスターイメージが使用されるようにします。ユーザーが次回ログオンすると、更新されたデスクトップが配信されます。この方法により、多数のユーザーに対する多くの変更を1回の操作で実行できます。

静的およびリモートPCアクセスマシンカタログのデスクトップは、Studioでは更新できません。サードパーティ製のソフトウェア配信ツールを使用して、個々のデスクトップまたはデスクトップのグループを更新します。Provisioning Servicesで作成されたマシンの場合は、vDiskを介してデスクトップを更新します。

更新する前に、マスターイメージのコピーまたはスナップショットを保存しておくことをお勧めします。データベースには、各マシンカタログで使用されたマスターイメージの履歴記録が保持されます。マスターイメージの削除、移動、または名前変更は行わないでください。デスクトップの更新によりユーザーの操作に問題が発生した場合は、以前のバージョンのマスターイメージに戻して、ユーザーのダウンタイムを最小限に抑えることができます。

以下の点に注意してください。

- 追加するマシンの数に応じて十分なプロセッサ、メモリ、ストレージが仮想化ホスト上にあることを確認してください。
- 十分な数のActive Directoryコンピューターアカウントが使用可能であることを確認してください。既存のコンピューターアカウントを使用する場合、使用可能なアカウントの数により、追加できるマシンの数が制限されることに注意してください。
- 追加するマシン用にStudioでActive Directoryコンピューターアカウントを作成する場合は、適切なドメイン管理者のアクセス許可も必要です。

1. Studioのナビゲーションペインで [マシンカタログ] を選択します。
2. マシンカタログを選択し、[操作] ペインの [マシンの追加] を選択します。
3. 追加する仮想マシンの数を選択します。
4. 新しいActive Directoryアカウントを作成する場合（追加する仮想マシンに必要な数のアカウントがない場合など）は、以下の手順に従います。

- 作成するアカウントのドメインと場所を選択します。
- アカウント名前付けスキームを指定します。番号記号 (#) により、名前に追加される連番または文字とその位置が定義されます（名前の先頭に番号記号を配置することはできません）。たとえば、名前付けスキームとして「PC-Sales-##」を指定して [0~9] を選択すると、PC-Sales-01、PC-Sales-02、PC-Sales-03のコンピューターアカウント名が作成されます。

既存のActive Directoryアカウントを使用する場合は、次の操作を行います。

- アカウントを参照するか、[インポート] をクリックしてアカウント名の一覧のCSVファイルを指定します。追加するマシンに十分な数のアカウントをインポートする必要があります。
- これらのアカウントはStudioで管理されるため、すべてのアカウントのパスワードのリセットをStudioに許可するか、アカウントのパスワードを指定します（すべてのアカウントで同じパスワードを使用する必要があります）。

マシンの作成はバックグラウンドプロセスとして実行され、多くのマシンを追加する場合には時間がかかることがあります。Studioを終了してもマシンの作成処理は続行されます。

1. Studioのナビゲーションペインで [マシンカタログ] を選択します。

2. マシンカタログを選択し、[操作] ペインの [マシンカタログの編集] を選択します。
3. [電源管理] ページでは、リモートPCアクセスカタログの電源管理設定を変更したり電源管理接続を選択したりします。[組織単位] ページでは、組織単位を追加または削除します。  
[説明] ページでは、マシンカタログの説明を変更します。

1. Studioのナビゲーションペインで [マシンカタログ] を選択します。
2. マシンカタログを選択し、[操作] ペインの [マシンカタログの名前を変更] を選択します。
3. 新しい名前を入力します。

マシンカタログを削除する前に、以下の点について確認してください。

- すべてのユーザーがログオフしており、実行中の切断セッションがないこと。
- カタログ内のすべてのマシンのメンテナンスモードがオンで、すべてのマシンがシャットダウンしていること。
- そのカタログがデリバリーグループに関連付けられていないこと。

1. Studioのナビゲーションペインで [マシンカタログ] を選択します。
2. マシンカタログを選択し、[操作] ペインの [マシンカタログの削除] を選択します。
3. カタログ内のマシンを削除するかどうかを指定します。マシンを削除する場合は、関連付けられているコンピューターアカウントをActive Directoryに残すか、無効にするか、削除するかを指定します。

マシンをカタログから削除すると、ユーザーはそのマシンにアクセスできなくなります。マシンを削除する前に、以下の点について確認してください。

- マシン上に重要なユーザーデータがなく、データがある場合はバックアップ済みであること。
- すべてのユーザーがログオフしていること。メンテナンスモードをオンにすると、ユーザーはマシンに接続できなくなります。
- デスクトップの電源がオンになっていたり一時停止状態になっていたりしないこと。

1. Studioのナビゲーションペインで [マシンカタログ] を選択します。
2. マシンカタログを選択し、[操作] ペインの [マシンの表示] を選択します。
3. 1台または複数のマシンを選択し、[操作] ペインの [メンテナンスモードをオンにする] をクリックします。
4. [操作] ペインの [削除] を選択します。
5. マシンを削除するかどうかを選択します。マシンを削除する場合は、関連付けられているActive Directoryコンピューターアカウントの処理を選択します。

| マシンカタログでの処理 | Active Directoryでの処理 |
|-------------|----------------------|
| そのまま        | 変更しない                |
| 削除          | 削除しない                |
| 削除          | 無効化                  |
| 削除          | 削除                   |

マシンカタログのActive Directoryアカウントについて、次の操作を行えます。

- デスクトップOSマシンカタログまたはサーバーOSマシンカタログからActive Directoryコンピューターアカウントを削除して未使用のマシンアカウントを解放する。解放したアカウントは、ほかのマシンで使用可能になります。
- カタログに追加するマシン用のコンピューターアカウントを追加しておく。

1. Studioのナビゲーションペインで [マシンカタログ] を選択します。
2. マシンカタログを選択し、 [操作] ペインの [Active Directoryアカウント管理] を選択します。
3. 必要に応じてコンピューターアカウントを追加または削除します。
  - アカウントを追加する場合は、すべてのアカウントのパスワードをリセットするか同じパスワードを指定するかを選択します。アカウントの現在のパスワードがわからない場合は、すべてのアカウントのパスワードをリセットするオプションを選択します。パスワードをリセットするための権限が必要です。パスワードを指定する場合は、アカウントのインポート時にパスワードが変更されます。
  - アカウントを削除する場合は、そのアカウントをActive Directory内で保持するか、無効にするか、または削除するかを選択します。

マシンカタログのマスターイメージを更新して、そのマスターイメージから作成されたマシンのすべてのデスクトップとアプリケーションに変更を適用できます。共通の要素を単一のマスターイメージで管理できるため、Windowsの更新プログラムご適用や構成の変更など、システムレベルの変更を多くのマシンにすばやく展開できます。

新しいマスターイメージまたは更新したマスターイメージをホスト上で準備およびテスト ([マシンカタログの作成](#) 参照) してから、そのマスターイメージが使用されるようにマシンカタログを変更します。

以下の点に注意してください。

- 更新する前に、マスターイメージのコピーまたはスナップショットを保存しておくことをお勧めします。データベースには、各マシンカタログで使用されたマスターイメージの履歴記録が保持されます。デスクトップの更新によりユーザーの操作に問題が発生した場合は、以前のバージョンのマスターイメージに戻して、ユーザーのダウンタイムを最小限に抑えることができます。マスターイメージの削除、移動、または名前変更は行わないでください。これらの操作を行うと、マシンカタログを元に戻してマスターイメージを使用することができなくなります。Studioでスナップショットを作成することもできますが、ハイパーバイザー側の管理コンソールでスナップショットを作成し、それをStudioで選択することをお勧めします。これにより、スナップショットに自動生成される名前を付けるのではなく、わかりやすい名前と説明を指定できます。
- GPUの仮想化機能を使用したマスターイメージを更新する場合は、XenServerのXenCenterを使用する必要があります。
- Provisioning Servicesのマシンカタログで変更内容を反映させるには、新しいvDiskを公開する必要があります。詳しくは、Provisioning Servicesのドキュメントを参照してください。
- マスターイメージを更新したら、Studioで各マシンを再起動する必要があります。これにより、マスターイメージの変更がマシンに適用され、ユーザーが使用できるようになります。ただし、自動的にマシンが再起動する場合もあります。たとえば、ユーザーがデスクトップからログオフする場合などです。また、既存の再起動スケジュールに基づいて再起動することもあります。Studioを使用してマシンを再起動することもできます。

1. Studioのナビゲーションペインで [マシンカタログ] を選択します。
2. マシンカタログを選択し、 [操作] ペインの [マシンの更新] を選択します。
3. [マスターイメージ] ページで、ホストおよび更新したマスターイメージを選択します。
4. [ロールアウト方法] ページで、更新したマスターイメージをユーザーデスクトップに適用するタイミング (次回シャットダウン時または即時) を指定します。
  - 次回シャットダウン時に適用するオプションを選択した場合は、イメージの更新についてユーザーに通知メッセージを

送信できます。

- 即時に適用するオプションを選択した場合は、すべてのマシンを同時に再起動するか特定の期間で分散して再起動するかを選択できます。ユーザーへの通知メッセージは、マシンの再起動の1分、5分、または15分前に送信できます。

1. Studioのナビゲーションペインで [マシンカタログ] を選択します。
2. マシンカタログを選択し、 [操作] ペインの [マシン更新のロールバック] を選択します。
3. 元に戻したマスターイメージをどのようにユーザーのデスクトップに適用するかを指定します。詳しくは、前の説明を参照してください。

ロールバックは、復元が必要なデスクトップにのみ適用されます。たとえば、マスターイメージの更新時にログアウトしなかったユーザーなど、更新したマスターイメージが適用されていないデスクトップのユーザーは、メッセージを受信したり強制的にログオフされたりすることはありません。

マシン上のVDAを新しいバージョンにアップグレードした場合は、マシンカタログをアップグレードする必要があります。すべてのVDAを最新バージョンにアップグレードして、最新の機能を使用できるようにすることをお勧めします。

注：Windows XPまたはWindows Vistaマシンでは以前のバージョンのVDAを使用する必要があるため、最新の機能を使用することはできません。それらのマシンのWindowsオペレーティングシステムをアップグレードできない場合は、それらを個別のマシンカタログで管理することをお勧めします。詳しくは、「[Windows XPまたはWindows Vista上のVDA](#)」および「[VDAの混在環境のサポート](#)」を参照してください。

マシンカタログをアップグレードする前に、次の操作を行います。

- Provisioning Servicesを使用している場合は、Provisioning ServicesコンソールでVDAをアップグレードします。
- アップグレードしたマシンを起動します。これにより、マシンがControllerに登録されます。このときに、そのマシンカタログ内のマシンについてアップグレードが必要かどうかStudioによりチェックされます。

1. Studioのナビゲーションペインで [マシンカタログ] を選択します。
2. マシンカタログを選択します。下ペインの [詳細] タブにバージョン情報が表示されます。
3. [カタログのアップグレード] を選択します。
  - Studioによりアップグレードが必要なことが検出されると、メッセージが表示されます。画面の指示に従って操作します。
  - アップグレードできないマシンがある場合は、その理由を説明するメッセージが示されます。すべてのマシンを適切に動作させるため、マシンカタログをアップグレードする前にマシンの問題を解決しておくことをお勧めします。

Provisioning Servicesを使用している場合は、マシンカタログのアップグレードを元に戻す前に、Provisioning ServicesコンソールでVDAバージョンを変更します。

1. Studioのナビゲーションペインで [マシンカタログ] を選択します。
2. マシンカタログを選択します。下ペインの [詳細] タブにバージョン情報が表示されます。
3. [元に戻す] を選択して、画面の指示に従います。

# デリバリーグループ

Jan 04, 2017

デリバリーグループは、いくつかのマシナカタログから選択したマシンをグループ化したものです。デリバリーグループでは、それらのマシンを使用できるユーザーと、そのユーザーに提供するアプリケーションを指定します。

まず、デリバリーグループを作成します。その後で、必要に応じて設定内容を変更します。

デリバリーグループを作成するには、次の手順に従います。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. [操作] ペインの [デリバリーグループの作成] を選択します。ウィザードの指示に従って、以下の操作を行います。

マシナカタログを選択して、そのマシナカタログから使用するマシンの数を指定します。

- マシナカタログに未使用のマシンが残っていない場合、そのカタログを選択することはできません。
- 同じマシナカタログを複数のデリバリーグループで選択することができます。ただし、同じマシンを複数のデリバリーグループで使用することはできません。
- 1つのデリバリーグループで複数のマシナカタログを使用できますが、これらのマシナカタログに同じ種類のマシン（サーバーOS、デスクトップOS、リモートPCアクセス）が含まれている必要があります。つまり、異なる種類のマシンをデリバリーグループやマシナカタログに混在させることはできません。
- 同様に、静的デスクトップ用に構成されたマシナカタログのデスクトップOSマシンと、ランダムデスクトップ用に構成されたマシナカタログのマシンが混在するデリバリーグループを作成することはできません。
- リモートPCアクセスマシナカタログの各マシンは自動的にデリバリーグループに関連付けられます。

配信の種類として、このデリバリーグループで提供するリソース（デスクトップのみ、アプリケーションのみ、またはデスクトップとアプリケーションの両方）を指定します。静的デスクトップOSマシンを使用するデリバリーグループでは、デスクトップとアプリケーションの両方を提供することはできません。

このデリバリーグループで配信されるアプリケーションやデスクトップを使用できるユーザーおよびユーザーグループを指します。

ユーザーには、認証が必要なユーザーと認証が不要なユーザーの2種類があります（認証が不要なユーザーは「匿名ユーザー」とも呼ばれます）。いずれか一方または両方の種類のユーザーを構成できます。

- **認証が必要なユーザー** — 特定のアカウント名で指定するユーザーおよびグループメンバーは、アプリケーションおよびデスクトップにアクセスするときにStoreFrontまたはCitrix Receiverで資格情報（スマートカード、またはユーザー名とパスワードなど）による認証が必要です。
- **認証が不要なユーザー（匿名ユーザー）** — サーバーOSマシン用のデリバリーグループでは、チェックボックスをオンにすることで、StoreFrontまたはCitrix Receiverでの認証が不要な匿名アクセスが可能になります。たとえば、キオスクからアプリケーションにアクセスする場合、アプリケーション自体での認証が必要な場合がありますが、Citrixのアクセスポータルやツールでは資格情報が要求されません。VDAをインストールすると、匿名のユーザーグループが作成されます。
- 認証が不要なユーザーのアクセスを許可するには、デリバリーグループの各マシンにVDA for Windows Server OS (Version 7.6以降) がインストールされている必要があります。認証が不要なユーザーのアクセスを有効にする場合は、認証が不要なStoreFrontストアを作成しておく必要があります。

- 認証が不要なユーザーアカウントはセッション開始時にオンデマンドで作成され、Anon<XYZ> (<XYZ>は一意の3桁の値) という名前が付きます。
- 認証が不要なユーザーのセッションにはデフォルトで10分のアイドルタイムアウトが設定され、セッションを切断すると自動的にログオフされます。切断セッションへの再接続、デバイス間のローミング、およびワークスペースコントロールはサポートされません。

次の表は、ウィザードの [ユーザー] ページで管理者が設定する内容を示しています。

| アクセスを許可するユーザー           | ユーザーおよびユーザーグループを追加/割り当てるかどうか | [認証が不要な (匿名) ユーザーのアクセスを許可する] チェックボックスをオンにするかどうか |
|-------------------------|------------------------------|-------------------------------------------------|
| 認証が必要なユーザーのみ            | はい                           | いいえ                                             |
| 認証が不要なユーザーのみ            | いいえ                          | はい                                              |
| 認証が必要なユーザーおよび認証が不要なユーザー | はい                           | はい                                              |

デスクトップOSマシン用のデスクトップグループでは、デリバリーグループを作成した後でユーザーデータ (ユーザーの一覧) をインポートできます。「[ユーザー一覧のインポートまたはエクスポート](#)」を参照してください。

マスターイメージから作成されたマシン、マシンカタログのテンプレート、またはApp-V公開サーバー上で検出されたアプリケーションが一覧に表示されます。このデリバリーグループに追加するアプリケーションを1つまたは複数選択します。

検出されていないアプリケーションを一覧に追加 (作成) することもできます。この場合、実行可能ファイルのパス、作業ディレクトリ、コマンドライン引数 (オプション)、管理者およびユーザー用の表示名を入力する必要があります。

検出済みのアプリケーションのプロパティを変更することもできます。詳しくは、「[アプリケーションプロパティの変更](#)」を参照してください。

リモートPCアクセス用のデリバリーグループにアプリケーションを追加することはできません。

アプリケーションをデリバリーグループに追加すると、デフォルトで「アプリケーション」という名前のフォルダー内に表示されます。Studioでフォルダーを使用すると、多数のアプリケーションをグループ化して管理できます。アプリケーションを追加するときに別のフォルダーを指定することもできますが、後でフォルダーを指定する方がより簡単に管理できます。詳しくは、「[アプリケーションフォルダーの管理](#)」を参照してください。

2つのアプリケーションを同じ名前と同じユーザーに公開する場合は、Studioで[アプリケーション名 (ユーザー用)] ボックスに別の名前を入力します。これを行わないと、ユーザーのReceiverに同じ名前が2つ表示されます。

StoreFrontのURLを選択または追加します。このURLはデリバリーグループの各マシンにインストールされたCitrix Receiverで使用されます (ユーザーがこのマシンにアクセスするときに使用するURLではありません)。StoreFrontサーバーのアドレスは、後で指定することもできます。これを行うには、ナビゲーションペインで [構成] > [StoreFront] の順に選択します。StoreFront Serverを追加する場合、'/Discovery'をURLの最後に追加します。

# 設定

Jan 04, 2017

以下のドキュメントでは、デリバリーグループの各項目の設定を構成および管理する方法について説明しています。

- [アプリケーション](#)
- [マシン](#)
- [リモートPCアクセス](#)
- [セッション](#)
- [ユーザー](#)

ここでは、上記のドキュメントに記述されていない設定について説明します。

アプリケーションのみ、またはデスクトップおよびアプリケーションのデリバリーグループをデスクトップのみのデリバリーグループに変更する前に、デリバリーグループからすべてのアプリケーションを削除します。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、 [操作] ペインの [デリバリーグループの編集] を選択します。
3. [配信の種類] ページで、デリバリーグループを変更させる配信の種類を選択します。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、 [操作] ペインの [デリバリーグループの編集] を選択します。
3. [基本設定] ページでは、以下の内容を変更できます。

| 設定                       | 説明                                                                                                                                                                                       |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 説明                       | StoreFrontでユーザーに表示される説明です。                                                                                                                                                               |
| [デリバリーグループの有効化] チェックボックス | このデリバリーグループを有効にするかどうかを設定します。                                                                                                                                                             |
| ユーザーごとのデスクトップ数           | (デスクトップOSマシンのみ) ユーザーが同時にアクティブにできる共有デスクトップの最大数です。初回使用時割り当てのデスクトップグループでは、この値によってユーザーが自分に割り当てることができるデスクトップ数が指定されます。                                                                         |
| タイムゾーン                   |                                                                                                                                                                                          |
| Secure ICAを有効にする         | デリバリーグループのマシンとの通信をSecureICAで保護します。これにより、ICAプロトコルが暗号化されます (デフォルトレベルは128ビットです。このレベルはSDKを使用して変更できます)。公共のネットワークが使用される環境では、SSL/TLSなどの暗号化方法を追加することをお勧めします。また、SecureICAでは、メッセージの整合性チェックが行われません。 |



デリバリーグループのアップグレードは、マシン上のVDAをアップグレードしてから行ってください。

注：VDAをアップグレードせずに使用を継続すると、新しい製品機能を使用できなくなる場合があります。詳しくは、[環境のアップグレード](#)」を参照してください。

デリバリーグループのアップグレードを開始する前に、以下の操作を行います。

- Provisioning Servicesを使用する環境では、Provisioning ServicesコンソールでVDAのバージョンをアップグレードします。
- 新しいVDAがインストールされているマシンを起動して、Controllerに登録します。この処理によって、デリバリーグループで必要なアップグレードがStudioで特定されます。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、 [操作] ペインの [デリバリーグループのアップグレード] を選択します。

アップグレードできないマシンがある場合は、そのマシンとアップグレードできない理由が表示されます。この場合はアップグレードをキャンセルし、マシンの問題を解決してから、デリバリーグループのアップグレードを再度開始できます。

デリバリーグループをアップグレードした後でマシンを元の状態に戻すには、デリバリーグループを選択し、 [操作] ペインの [元に戻す] を選択します。

# マシン

Jan 19, 2017

特に明記されている場合を除き、デリバリーグループのすべての種類（サーバーOS、デスクトップOS、およびリモートPCアクセス）で以下の手順がサポートされます。

注：ここで説明する内容は、リモートPCアクセスマシンではサポートされません。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、[操作] ペインの [マシンの表示] を選択します。
3. マシンを選択し、[操作] ペインで以下のいずれかを選択します（マシンの状態によっては選択できないオプションもあります）。
  - 強制シャットダウン — マシンの電源を強制的に切って、マシン一覧を更新します。
  - 再起動 — マシンのオペレーティングシステムに再起動を要求します。オペレーティングシステムで再起動を実行できない場合、マシンの状態は変更されません。
  - 一時停止 — マシンをシャットダウンすることなく一時的に停止して、マシン一覧を更新します。
  - シャットダウン — マシンのオペレーティングシステムにシャットダウンを要求します。

マシンが10分以内にシャットダウンしない場合、電源が切れ、強制的にシャットダウンされます。シャットダウン中にWindowsが更新のインストールを開始すると、更新が完了する前にマシンの電源が切れる危険性があります。

注：セッション中はデスクトップOSマシンのユーザーにシャットダウンの選択を禁止することをお勧めします。詳しくは、Microsoftのポリシーのドキュメントを参照してください。

注：電源を管理できるのは、仮想デスクトップOSマシンのみです。物理マシンの電源を管理することはできません（リモートPCアクセスマシンを含む）。GPU機能が有効なデスクトップOSマシンは一時停止できないため、電源を切ることはできません。サーバーOSマシンについては、「再起動スケジュールの作成」を参照してください。

マシンは、以下のいずれかの状態になります。

| Delivery Group | State                 |
|----------------|-----------------------|
| ランダム           | ランダムに割り当てられ、使用中       |
|                | 未割り当て、未接続             |
| 静的（割り当て済み）     | 永続的に割り当てられ、使用中        |
|                | 永続的に割り当てられ、未接続（準備は完了） |
|                | 未割り当て、未接続             |

通常、静的なデリバリーグループには、永続的に割り当てられたマシンと未割り当てマシンの両方が含まれています。最初、すべてのマシンは未割り当て状態です（デリバリーグループ作成時に手で割り当てられたマシンを除く）。ユーザーが接続すると、マシンが永続的に割り当てられます。静的なデリバリーグループでは未割り当てマシンの電源を完全に管理できませんが、永続的に割り当てられたマシンでは一部の電源管理のみを実行できます。

- **プールおよびバッファ** — 静的なデリバリーグループの未割り当てマシン、およびランダムなデリバリーグループの場合、「プール」は電源が入った未割り当て（または一時的に割り当てられた）マシンのセットを指し、ユーザーがログオンすると直ちにマシンが割り当てられます。プールサイズ（電源が入った状態のマシンの数）は時刻によって構成できません。静的なデリバリーグループでは、SDKを使用してプールを構成します。「バッファ」は追加の未割り当てマシンの「待機」セットを指し、プール内のマシンの数がデリバリーグループのサイズに対する割合により設定されたしきい値を下回ると、バッファのマシンの電源がオンになります。大規模なデリバリーグループの場合、このしきい値により多数のマシンの電源がオンになることがあります。このため、デリバリーグループのサイズを小さくするか、SDKを使ってデフォルトのバッファサイズを調節してください。
- **電源状態タイマー** — 電源状態タイマーを使用して、ユーザーが切断してから一定の時間が経過したマシンを一時停止にすることができます。たとえば、業務時間終了後にユーザーが切断してから10分が経過したマシンを自動的に一時停止状態にできます。ランダムなマシンまたはPersonal vDisksを使用しているマシンは、ユーザーがログオフすると自動的にシャットダウンします（SDKでデリバリーグループのShutdownDesktopsAfterUseプロパティを構成している場合を除く）。平日と週末、ピーク期間とオフピーク期間のタイマーを構成できます。
- **永続的に割り当てられたマシンの部分的な電源管理** — 永続的に割り当てられたマシンでは、電源状態タイマーを設定することはできませんが、プールまたはバッファを設定することはできません。各ピーク時間の開始時にマシンの電源がオンになり、各オフピーク時間の開始時に電源がオフになります。このため、未割り当てマシンの場合とは異なり、使用中のマシンを補うためのマシンの数を詳細には制御できません。

仮想デスクトップOSマシンの電源を管理するには、次の手順に従います。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、[操作] ペインの [デリバリーグループの編集] を選択します。
3. [電源管理] ページの [マシンの電源管理] ボックスで [1 - 平日] を選択します（平日は、デフォルトで月曜日から金曜日です）。
4. ランダムなデリバリーグループの場合、[電源をオンするマシン] で [編集] をクリックして、平日のプールサイズを指定します。次に、電源をオンにするマシンの数を選択します。
5. [ピーク時] で、平日のピーク時間とオフピーク時間を設定します。
6. 平日のピーク時間およびオフピーク時間の電源状態タイマーを設定します。
  - [ピーク時の電源管理] > [切断時] で、ユーザーが切断してからマシンを一時停止状態にするまでの時間（分）を指定して [一時停止] を選択します。
  - [オフピーク時の電源管理] > [切断時] で、ユーザーがログオフしてからマシンの電源をオフにするまでの時間を指定して [シャットダウン] を選択します。このタイマーはランダムマシンのデリバリーグループでは使用できません。
8. [マシンの電源管理] ボックスで [2 - 週末] を選択し、週末のピーク時間と電源状態タイマーを構成します。

SDKを使用すると、以下の設定が可能です。

- 電源状態タイマーの設定に基づいてマシンを（一時停止ではなく）シャットダウンする場合や、ユーザーの（切断時ではなく）ログオフ時にタイマーが起算されるように設定する。
- デフォルトの平日と週末の定義を変更する。

注：再起動スケジュールはサーバーOSマシンでのみ使用できます。デスクトップOSマシンについては、[マシンの電源管理](#)を参照してください。

再起動スケジュールを構成するには、次の手順に従います。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、[操作] ペインの [デリバリーグループの編集] を選択します。
3. [再起動のスケジュール] ページで、以下の項目を設定します。

- [マシンの再起動] で、マシンを再起動する曜日を指定します。
- [最初のグループの再起動時刻] で、最初のサーバーの再起動を開始する時刻（24時間形式）を指定します。
- [後続グループの再起動間隔] で、すべてのサーバーを一度に再起動するか、間隔を空けて各サーバーを再起動する場合はその間隔を指定します。  
たとえば、デリバリーグループに5つのサーバーがある場合、[最初のグループの再起動時刻] で13:00（午後1:00）を指定して [後続グループの再起動間隔] で [1時間] を指定すると、5つのサーバーの再起動が1時間内に分散されるため、12分間隔（60÷5）で再起動されます。このため、午後1:00、1:12、1:24、1:36、および1:48にサーバーが再起動されます。これにより、指定した時間内（1時間）にすべてのサーバーの再起動が完了します。
- ユーザーをログオフする前に通知メッセージを送信するかどうか、送信する場合はそのタイミングを選択し、メッセージの内容を入力します。通知メッセージは、上の例で計算された各サーバーの再起動時刻の前に送信されます。

Studioでは、マシンの電源を自動的に投入したりシャットダウンしたりすることはできません。

一時的に新しい接続を停止する必要がある場合は、デリバリーグループの1台またはすべてのマシンに対してメンテナンスモードを有効にすることができます。パッチを適用したりメンテナンスツールを使用したりする場合は、メンテナンスモードを有効にしてから実行することをお勧めします。

- メンテナンスモードのサーバーOSマシンでは、既存のセッションに接続することはできますが、新しいセッションを開始することはできません。
- メンテナンスモードのデスクトップOSマシン（またはリモートPCアクセスを使用しているPC）では、新しいセッションを開始することも既存のセッションに再接続することもできません。実行中の接続は、ユーザーが切断またはログオフするまでは保持されます。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択します。
3. デリバリーグループのすべてのマシンをメンテナンスモードにするには、[操作] ペインの [メンテナンスモードをオンにする] を選択します。

特定のマシンをメンテナンスモードにするには、次の手順に従います。

1. [操作] ペインの [マシンの表示] を選択します。
2. マシンを選択し、[操作] ペインの [メンテナンスモードをオンにする] を選択します。
4. 特定のマシンまたはデリバリーグループのすべてのマシンのメンテナンスモードを解除するには、上記の手順に従って、[操作] ペインでは [メンテナンスモードをオフにする] を選択します。

Windowsリモートデスクトップ接続（RDC）の設定も、サーバーOSマシンをメンテナンスモードにするかどうかに影響します。次の状態のいずれかが発生すると、サーバーがメンテナンスモードになります。

- 上記の手順でサーバーの [メンテナンスモードをオンにする] が選択された。
- RDCが [このコンピューターへの接続を許可しない] に設定された。
- RDCが [このコンピューターへの接続を許可しない] に設定されていない場合、リモートホスト構成のユーザーログオンモード設定が次のいずれかに設定された。
  - 再接続を許可するが、新しいログオンを許可しない
  - 再接続を許可するが、サーバーが再起動するまで新しいログオンを許可しない

注：マシンの割り当て先ユーザーを変更できるのはデスクトップOSマシンのみです。サーバーOSマシンやProvisioning Servicesで作成されたマシンは再割り当てできません。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択します。

3. 複数のマシンを再割り当てするには、次の手順に従います。
  1. [操作] ペインの [デリバリーグループの編集] を選択します。
  2. [マシン割り当て] (ユーザー割り当て) ページで、マシンを選択して新しいユーザーを指定します。
4. 特定のマシンを再割り当てするには、次の手順に従います。
  1. [操作] ペインの [マシンの表示] を選択します。
  2. マシンを選択して、[操作] ペインの [ユーザーの変更] を選択します。
  3. ユーザーを追加または削除します。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、[操作] ペインの [デリバリーグループの編集] を選択します。
3. [ユーザー設定] ページで、ユーザーごとのデスクトップ数を設定します。

注：タグはデスクトップOSマシンでのみ使用できます。

タグを使用して、マシン検索を絞り込んだり、マシンへのアクセスを制限したりできます。追加するタグの数や文字数に制限はありません。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、[操作] ペインの [マシンの表示] を選択します。
3. マシンを選択します。
4. タグを追加するには、[操作] ペインの [タグの追加] を選択し、1つまたは複数のタグをセミコロンで区切って入力します。  
既存のタグを変更または削除するには、[操作] ペインの [タグの編集] を選択し、必要な変更を行います。

注：負荷管理できるのはサーバーOSマシンのみです。

負荷管理機能では、測定されたサーバー負荷に基づいて最適なサーバーが選択されます。この選択は、以下の基準により行われます。

- **サーバーのメンテナンスモードの状態** — メンテナンスモードがオフのサーバーOSマシンだけが負荷分散の対象として選択されます。詳しくは、「[マシンへのユーザーの接続を禁止する \(メンテナンスモード\)](#)」を参照してください。
- **サーバー負荷指数** — サーバーOSマシンの配信サーバーの負荷に基づいて、そのサーバーがどれだけの接続を受け入れられるかが決定されます。サーバー負荷指数は、セッション数とパフォーマンス測定値 (CPU、ディスク、メモリ使用量など) で計算される負荷評価基準の組み合わせを指します。負荷評価基準は、ポリシーの負荷管理に関する設定項目で指定します。
  - Director、Studioの [検索] ノード、およびSDKを使用して負荷指数を監視できます。
  - Studioのデフォルトでは、[負荷指数] 列が表示されません。[負荷指数] 列を表示するには、マシンを選択し、列見出しを右クリックして [列の選択] を選択します。[マシン] の [負荷指数] チェックボックスをオンにします。
  - SDKではGet-BrokerMachineコマンドレットを使用します。  
[負荷指数] 列に値10000が表示される場合、そのサーバーが負荷限界状態であることを示しています。ほかに使用可能なサーバーがない場合は、ユーザーがセッションを起動したときに、デスクトップまたはアプリケーションを使用できないという内容のメッセージが表示されます。
- **同時ログオンレランスのポリシー設定** — サーバーが同時に処理できるログオン要求の最大数です。この設定項目は、Version 7.5より前のXenAppの「負荷調整」に相当します。

すべてのサーバーが同時ログオンレランスの設定値に達した場合、それ以降のログオン要求は保留中のログオン数が最も少ないサーバーに割り当てられます。同時ログオンレランスの設定値に達しないサーバーがいくつか存在する場合は、負荷

数が最小のサーバーにログオン要求が割り当てられます。

詳しくは、「  
—ポリシー設定リファレンス  
」を参照してください。

デリバリーグループからマシンを削除しても、そのデリバリーグループで使用するマシンカタログからは削除されません。そのため、そのマシンをほかのデリバリーグループで使用することができます。

マシンを削除する前に、マシンをシャットダウンする必要があります。デリバリーグループから削除せずにマシンを一時的に使用できなくする場合は、そのマシンをメンテナンスモードにしてからシャットダウンしてください。

マシンには個人データが保存されている可能性があります。このため、そのマシンを別のユーザーに割り当てる場合は注意が必要です。マシンをイメージから再作成することをお勧めします。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、[操作] ペインの [マシンの表示] を選択します。
3. マシンがシャットダウン状態であることを確認します。
4. [操作] ペインの [デリバリーグループから削除] を選択します。

デリバリーグループでマシンへのアクセス制限を変更した場合、使用方法にかかわらず既存の設定より優先されます。マシンの追加方法

- 管理者のアクセスを制限する場合は、委任管理スコープを使用します。すべてのアプリケーションへのアクセスを許可するスコープや、特定のアプリケーションへのアクセスのみを許可するスコープを作成して管理者に割り当てることができます。詳しくは、委任管理のトピックを参照してください。
- ユーザーのアクセスを制限する場合は、NetScaler Gateway経由のユーザー接続を制御するSmartAccessポリシー式を使用します。
  1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
  2. デリバリーグループを選択して、[操作] ペインの [デリバリーグループの編集] を選択します。
  3. [アクセスポリシー] ページで、[NetScaler Gatewayを経由する接続] を選択します。
  4. NetScaler Gatewayを経由する特定の接続のみを許可するには、[次のフィルターのいずれかに一致する接続] チェックボックスをオンにします。次にNetScaler Gatewayサイトを定義して、接続を許可するユーザーを特定するSmartAccessポリシー式を追加、編集、または削除します。詳しくは、NetScaler Gatewayのドキュメントを参照してください。
- SDKで設定するアクセスポリシーの除外フィルターを使用してユーザーのアクセスを制限することもできます。アクセスポリシーはデリバリーグループに適用され、接続をより詳細に制御できます。たとえば、マシンへのアクセスを一部のユーザーに限定したり、特定のユーザーデバイスに限定したりできます。除外フィルターを使用するとアクセスポリシーをより詳細に定義できます。たとえば、セキュリティ上の理由により、一部のユーザーまたはデバイスからのアクセスを拒否できます。  
デフォルトでは、除外フィルターは無効になっています。

たとえば、社内ネットワークのサブネットにある教育ラボで、マシンを使用するユーザーにかかわらず教育ラボから特定のデリバリーグループへのアクセスを禁止する場合は、「Set-BrokerAccessPolicy -Name VPDesktops\_Direct -ExcludedClientIPFilterEnabled \$True -」コマンドを使用します。

ワイルドカード文字としてアスタリスク (\*) を使用し、同じポリシー式で始まるタグをすべて一致させることもできます。たとえば、タグ「VPDesktops\_Direct」が追加されたマシンと、タグ「VPDesktops\_Test」が追加されたマシンの両方をフィルターの対象にする場合は、Set-BrokerAccessPolicyスクリプトでタグとして「VPDesktops\_\*」を指定します。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、 [操作] ペインの [マシンの表示] を選択します。
3. マシンを選択して、 [操作] ペインの [ユーザーの更新] を選択します。
  - 別のマスターイメージを選択するには、 [マスターイメージ] を選択します。次にスナップショットを選択します。選択したスナップショットを展開すると、関連するマスターイメージが表示されます。
  - 変更内容を適用し、マシンのユーザーに通知するには、 [エンドユーザーへのロールアウト通知] を選択します。次に、以下を指定します。
    - マスターイメージをすぐに更新するのか次の起動時に更新するのかを指定します。
    - 再起動を分散させる時間を選択します。すべてのマシンを同時に再起動したり、特定の時間内に分散させたりできません。
    - ユーザーに再起動を通知するかどうか、通知するタイミング、および通知するメッセージを指定します。

# アプリケーション

Jan 04, 2017

デリバリーグループにアプリケーションを追加するには、次の手順に従います。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択します。
3. [操作] ペインの [アプリケーションの追加] を選択します。

マスターイメージから作成されたマシン、マシンカタログのテンプレート、またはApp-V公開サーバー上で検出されたアプリケーションが一覧に表示されます。このデリバリーグループに追加するアプリケーションを1つまたは複数選択します。

検出されていないアプリケーションを一覧に追加（作成）することもできます。この場合、実行可能ファイルのパス、作業ディレクトリ、コマンドライン引数（オプション）、管理者およびユーザー用の表示名を入力する必要があります。

検出済みのアプリケーションのプロパティを変更することもできます。詳しくは、後述の「アプリケーションプロパティの変更」を参照してください。

アプリケーションをデリバリーグループに追加すると、デフォルトで「アプリケーション」という名前のフォルダー内に表示されます。アプリケーションのフォルダーについて詳しくは、後述の「アプリケーションフォルダーの管理」を参照してください。

アプリケーションの複製、無効化、名前変更、タグ編集、削除を行うには、次の手順に従います。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. 中央ペインで [アプリケーション] タブを選択し、アプリケーションを選択します。
3. [操作] ペインで適切なタスクを選択します。

ヒント：

- アプリケーションを複製すると、自動的にアプリケーション名が設定され、元のアプリケーションに隣接して配置されます。
- アプリケーションを削除すると、デリバリーグループからアプリケーションが削除されます。ただし、マスターイメージのアプリケーションはアンインストールされません。
- アプリケーションを別のアプリケーションフォルダーに移動するには、後述の「アプリケーションフォルダーの管理」を参照してください。

アプリケーションのプロパティを変更するには、次の手順に従います。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. 中央ペインで [アプリケーション] タブを選択し、アプリケーションを選択します。
3. [操作] ペインの [プロパティ] を選択します。

以下の項目を表示および変更できます。

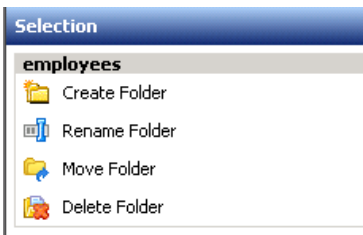
| 表示または変更するプロパティ | 選択するページ |
|----------------|---------|
| アプリケーション名      | 識別      |



| Receiverでの表示カテゴリ<br>表示または変更するプロパティ | 配信<br>選択するページ |
|------------------------------------|---------------|
| コマンドライン引数                          | 場所            |
| 説明                                 | 識別            |
| ファイル拡張子                            | ファイルタイプの関連付け  |
| ファイルタイプの関連付け                       | ファイルタイプの関連付け  |
| アイコン                               | 配信            |
| StoreFront用のキーワード                  | 識別            |
| 実行可能ファイルのパス                        | 場所            |
| ユーザーのデスクトップにショートカットを表示するかどうか       | 配信            |
| 特定のユーザーに表示するかどうか                   | 表示の制限         |
| 作業ディレクトリ                           | 場所            |

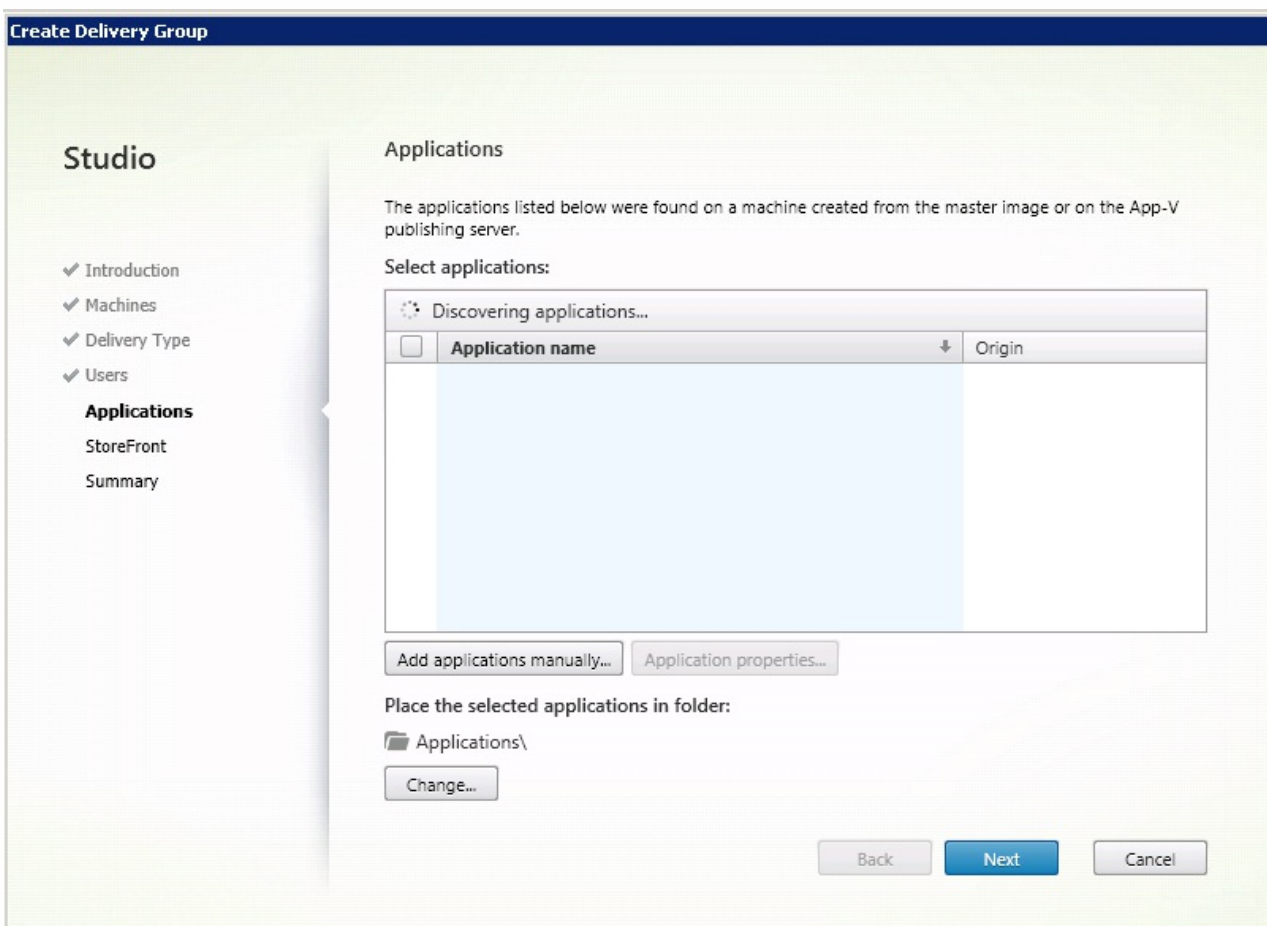
使用中のアプリケーションに変更内容を反映させるには、ユーザーがそのセッションからログオフする必要があります。

アプリケーションをデリバリーグループに追加すると、デフォルトで「  
—アプリケーション  
」という名前のフォルダー内に表示されます。マシンの追加方法



- 追加のフォルダーを作成して、アプリケーションをそのフォルダーに移動する。
  - フォルダーは5レベルまで入れ子にできます。
  - アプリケーションを含まない空のフォルダーを使用できます。
  - フォルダーは、移動したり作成時に別の場所を指定したりしない限り、アルファベット順に表示されます。
  - 親フォルダーが異なる限り、同じ名前の子フォルダーを作成できます。同様に、保存先フォルダーが異なる限り、同じ名前のアプリケーションを作成できます。
- フォルダーを同じレベルまたは異なるレベルに移動する。ドラッグアンドドロップによる移動が最も簡単です。
- 作成したフォルダーの名前を変更したり、作成したフォルダーを削除したりする。「アプリケーション」フォルダーの名前を変更したり、「アプリケーション」フォルダーを削除したりすることはできません。ただし、「アプリケーション」フォルダー内のすべてのアプリケーションを、作成済みの別のフォルダーに移動することは可能です。

アプリケーションのフォルダーは、デリバリーグループの作成時に作成することもできます。

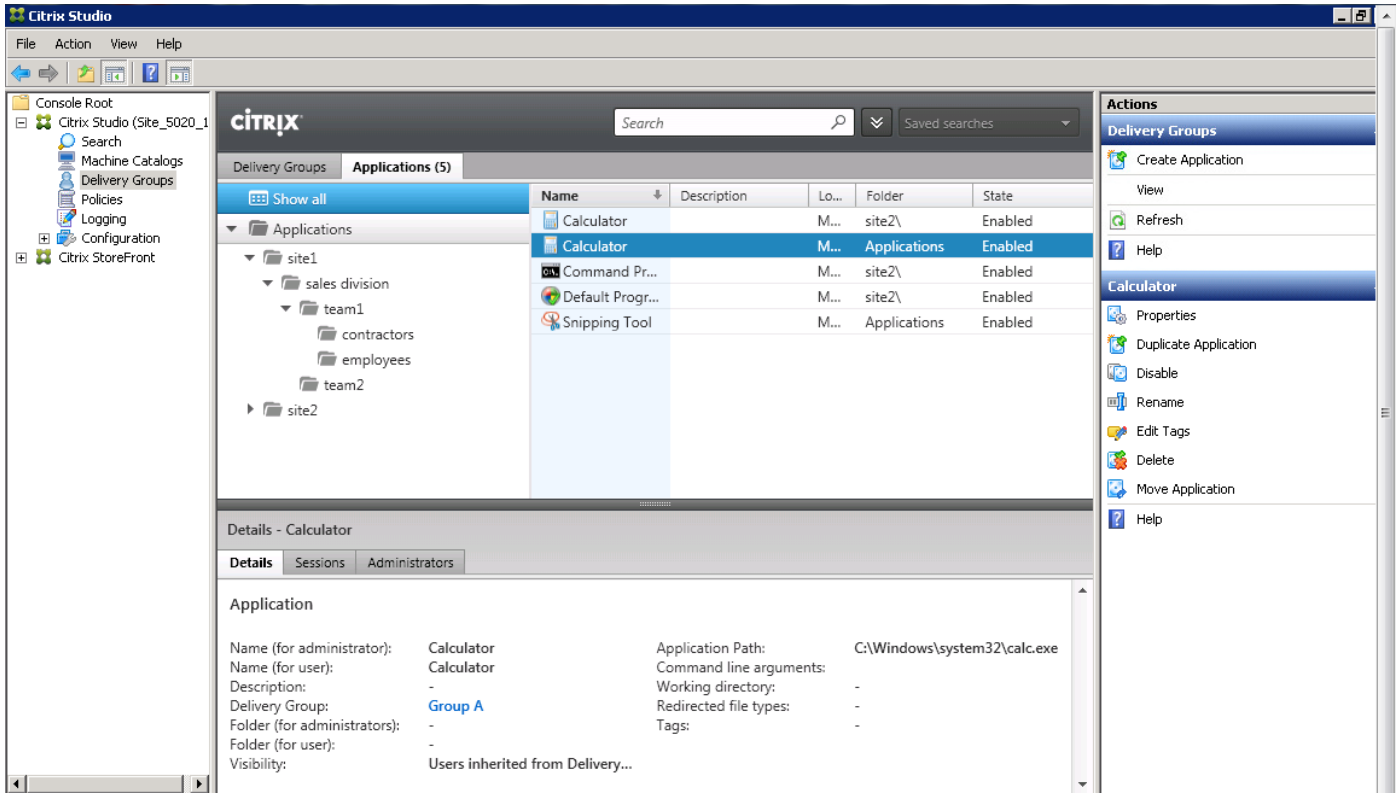


フォルダー内のアプリケーションを表示するには、[アプリケーションの表示] 権限が必要です。また、フォルダー内のアプリケーションを削除したり、フォルダー内のアプリケーション名を変更したり、アプリケーションが含まれるフォルダー

を削除したりするには、フォルダーに含まれるすべてのアプリケーションに対する [アプリケーションプロパティの編集] 権限が必要です。詳しくは、「[管理権限の委任](#)」を参照してください。

ヒント：以下の手順では、Studioの [操作] ペインを使用します。また、右クリックメニューやドラッグアンドドロップも使用できます。意図しない場所にフォルダーを作成または移動した場合は、正しい場所にドラッグアンドドロップできます。Studioのナビゲーションペインで [デリバリーグループ] を選択し、中央ペインで [アプリケーション] タブを選択します。

- すべてのフォルダー（サブフォルダーを除く）のアプリケーションを表示するには、[すべて表示] をクリックします。



- フォルダーを作成するには、次の手順に従います。
  1. フォルダーを最上位レベルに作成する（別のフォルダーのサブフォルダーにしない）場合は、最上位の「アプリケーション」フォルダーを選択します。「アプリケーション」フォルダー以外の既存のフォルダー内に新しいフォルダーを作成するには、その既存のフォルダーを選択します。
  2. [操作] ペインの [フォルダーの作成] を選択します。フォルダー名を1~64文字で入力します。スペースを使用できません。
- フォルダーを移動するには、次の手順に従います。
  1. フォルダーを選択して、[操作] ペインの [フォルダーの移動] を選択します。サブフォルダーを持つフォルダーを除き、一度に複数のフォルダーを移動することはできません。
  2. フォルダーを最上位レベルに移動する（別のフォルダーのサブフォルダーにしない）場合は、最上位の「アプリケーション」フォルダーを選択します。「アプリケーション」フォルダー以外の既存のフォルダー内にフォルダーを移動するには、その既存のフォルダーを選択します。
- フォルダー名を変更するには、名前を変更するフォルダーを選択し、[操作] ペインの [フォルダー名の変更] を選択します。新しいフォルダー名を1~64文字で入力します。
- フォルダーを削除するには、削除するフォルダーを選択し、[操作] ペインの [フォルダーの削除] を選択します。アプリケーションやサブフォルダーを含んでいるフォルダーを削除すると、それらのアプリケーションやサブフォルダーも削除されます。アプリケーションを削除すると、そのアプリケーションの割り当てがデリバリーグループから削除されます。マシンからアンインストールされることはありません。
- アプリケーションを別のフォルダー内に移動するには、1つまたは複数のアプリケーションを選択し、[操作] ペイン

の [アプリケーションの移動] を選択します。移動先のフォルダーを選択します。

デリバリーグループの作成ウィザードでアプリケーションをフォルダーに追加または移動するには、[アプリケーション] ページで1つまたは複数のアプリケーションを選択し、[変更] をクリックします。

- アプリケーションを既存のフォルダーに移動するには、移動先の既存のフォルダーを選択します。
- アプリケーションを新しいフォルダーに移動するには、次の手順に従います。
  - フォルダーを最上位レベルに作成する（別のフォルダーのサブフォルダーにしない）場合は、最上位の「アプリケーション」フォルダーを選択し、[新しいフォルダー] を選択します。新しいフォルダー名を1~64文字で入力します。スペースを使用できます。
  - 「アプリケーション」フォルダー以外の既存のフォルダー内に新しいフォルダーを作成するには、その既存のフォルダーを選択して、[新しいフォルダー] を選択します。新しいフォルダー名を1~64文字で入力します。スペースを使用できます。

# ユーザー

Jan 04, 2017

ユーザーには、認証が必要なユーザーと認証が不要なユーザーの2種類があります（認証が不要なユーザーは「匿名ユーザー」とも呼ばれます）。いずれか一方または両方の種類のユーザーを構成できます。

- **認証が必要なユーザー** — 特定のアカウント名で指定するユーザーおよびグループメンバーは、アプリケーションおよびデスクトップにアクセスするときにStoreFrontまたはCitrix Receiverで資格情報（スマートカード、またはユーザー名とパスワードなど）による認証が必要です。
- **認証が不要なユーザー（匿名ユーザー）** — サーバーOSマシン用のデリバリーグループでは、チェックボックスをオンにすることで、StoreFrontまたはCitrix Receiverでの認証が不要な匿名アクセスが可能になります。たとえば、キオスクからアプリケーションにアクセスする場合、アプリケーション自体での認証が必要な場合がありますが、Citrixのアクセスポータルやツールでは資格情報が要求されません。VDAをインストールすると、匿名のユーザーグループが作成されます。
  - 認証が不要なユーザーのアクセスを許可するには、デリバリーグループの各マシンにVDA for Windows Server OS (Version 7.6以降) がインストールされている必要があります。認証が不要なユーザーのアクセスを有効にする場合は、認証が不要なStoreFrontストアを作成しておく必要があります。
  - 認証が不要なユーザーアカウントはセッション開始時にオンデマンドで作成され、Anon<XYZ> (<XYZ>は一意的3桁の値) という名前が付きます。
  - 認証が不要なユーザーのセッションにはデフォルトで10分のアイドルタイムアウトが設定され、セッションを切断すると自動的にログオフされます。切断セッションへの再接続、デバイス間のローミング、およびワークスペースコントロールはサポートされません。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、[操作] ペインの [デリバリーグループの編集] を選択します。
3. 次の表は、ウィザードの [ユーザー] ページで管理者が設定する内容を示しています。

| アクセスを許可するユーザー           | ユーザーおよびユーザーグループを追加/割り当てるかどうか | [認証が不要な（匿名）ユーザーのアクセスを許可する] チェックボックスをオンにするかどうか |
|-------------------------|------------------------------|-----------------------------------------------|
| 認証が必要なユーザーのみ            | はい                           | いいえ                                           |
| 認証が不要なユーザーのみ            | いいえ                          | はい                                            |
| 認証が必要なユーザーおよび認証が不要なユーザー | はい                           | はい                                            |

デスクトップOSマシン用のデスクトップグループでは、デリバリーグループを作成した後でユーザーデータ（ユーザーの一覧）をインポートできます。次の「[ユーザー一覧のインポートまたはエクスポート](#)」を参照してください。

物理デスクトップOSマシン用のデリバリーグループでは、デリバリーグループを作成した後でCSVファイルからユーザー情報をインポートできます。ユーザー情報をCSVファイルにエクスポートすることもできます。以前の製品バージョンでのユーザー情報をCSVファイルに含めることもできます。

このCSVファイルの最初の行は、コンマで区切られた列見出し（順不同）でADComputerAccount、AssignedUser、VirtualMachine、およびHostIdが含まれます。以降の行には、コンマで区切られたデータが含まれます。

ADComputerAccountエントリは、共通名、IPアドレス、識別名、またはドメインとコンピューター名のペアです。

ユーザー情報をインポートまたはエクスポートするには、次の手順に従います。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、 [操作] ペインの [デリバリーグループの編集] を選択します。
3. [マシン割り当て] ページで、 [一覧のインポート] または [一覧のエクスポート] を選択し、ファイルの場所を参照します。

# セッション

Jan 04, 2017

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、 [操作] ペインの [マシンの表示] を選択します。
3. ユーザーをセッションからログオフするには、セッションまたはデスクトップを選択し、 [操作] ペインの [ログオフ] を選択します。セッションが閉じて、ほかのユーザーがそのマシンを使用できるようになります（そのマシンが特定のユーザーに割り当てられてない場合）。

セッションを切断するには、セッションまたはデスクトップを選択し、 [操作] ペインの [切断] を選択します。ユーザーのアプリケーションはそのまま実行され、マシンはそのユーザーに割り当てられたままになります。ユーザーは同じマシンに再接続できます。

ユーザーにメッセージを送信するには、セッション、マシン、またはユーザーを選択し、 [操作] ペインの [メッセージの送信] を選択します。メッセージを入力します。

デスクトップOSマシンでは、電源状態タイマーを使用して、ユーザーが切断してから一定の時間が経過したマシンを一時停止にしたりシャットダウンしたりすることができます。詳しくは、「[マシンの電源管理](#)」を参照してください。

注：これらの機能はサーバーOSマシンでのみサポートされます。

セッションの事前起動機能とセッション残留機能を使用すると、セッションが要求される前にセッションを開始したり（セッションの事前起動）、ユーザーがすべてのアプリケーションを閉じた後もアプリケーションセッションをアクティブな状態で保持したり（セッション残留）できます。これにより、ユーザーがアプリケーションにすばやくアクセスできるようになります。

デフォルトでは、セッションの事前起動とセッション残留は無効になっています。セッションはユーザーがアプリケーションを開始すると開始（起動）され、セッションで開いていた最後のアプリケーションを閉じるまでアクティブな状態で保持されます。

以下の点に注意してください。

- これらの機能を使用するには、デリバリーグループでアプリケーションが配信されている必要があります。また、マシンでVDA for Server OS 7.6以降が動作している必要があります。
- セッション事前起動は、Citrix Receiver for Windowsを使用している場合にのみサポートされます。セッション残留は、Citrix Receiver for WindowsおよびReceiver for Webを使用している場合にサポートされます。追加の構成は必要ありません。詳しくは、適切なバージョンのReceiver for WindowsのeDocsドキュメントで「セッションの事前起動」を検索して

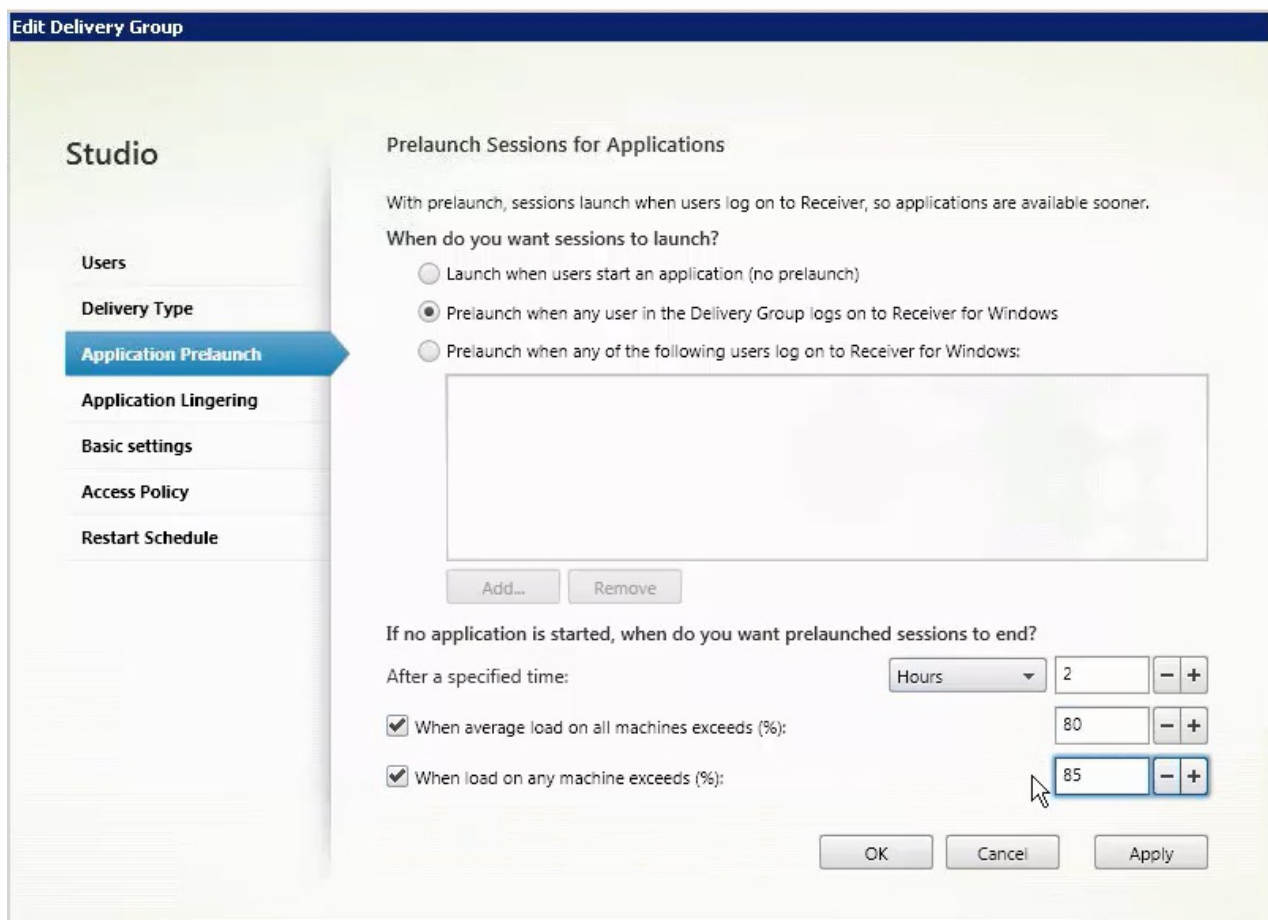
ください。

注：Receiver for HTML5はサポートされません。

- セッションの事前起動を使用する場合は、以下の点に注意してください。
  - 管理側の設定とは関係なく、エンドユーザーのマシンが「一時停止」または「休止状態」モードになると、事前起動は機能しません。
  - 事前起動はエンドユーザーがマシンやセッションをロックする限り機能しますが、エンドユーザーがCitrix Receiverからログオフすると、セッションが終了し、事前起動は適用されません。
- 事前起動セッションと残留セッションは、接続されている間のみライセンスを消費します。使用されない事前起動セッションと残留セッションは、デフォルトで15分後に切断されます。この値はPowerShell (New/Set-BrokerSessionPreLaunchコマンドレット) で構成できます。
- これらの機能が相互に補完し合うよう調整するには、ユーザーの使用状況を監視して慎重に計画することが重要です。最適に構成することで、ライセンス消費やリソース割り当ての効率化とユーザーの利便性を両立させることができます。
- Receiver側では、セッションの事前起動を有効にする時間帯を構成できます。

セッションの事前起動を有効にするには、次の手順に従います。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、 [操作] ペインの [デリバリーグループの編集] を選択します。
3. [アプリケーションの事前起動] ページで、セッションを起動するタイミングを選択します。
  - アプリケーションの起動時にセッションを起動する。これがデフォルトの設定で、セッションの事前起動は無効になります。
  - デリバリーグループ内のすべてのユーザーでReceiver for Windowsログオン時に事前起動する。
  - 以下のユーザーでのみReceiver for Windowsログオン時に事前起動する。このオプションを選択する場合は、ユーザーまたはユーザーグループを一覧に追加してください。





4. 事前起動セッションは、ユーザーがアプリケーションを起動すると通常のセッションに置き換わります。ユーザーがアプリケーションを起動しない場合（事前起動セッションが使用されない場合）、以下の設定に従って事前起動セッションが終了します。これらの設定について詳しくは、後述の「

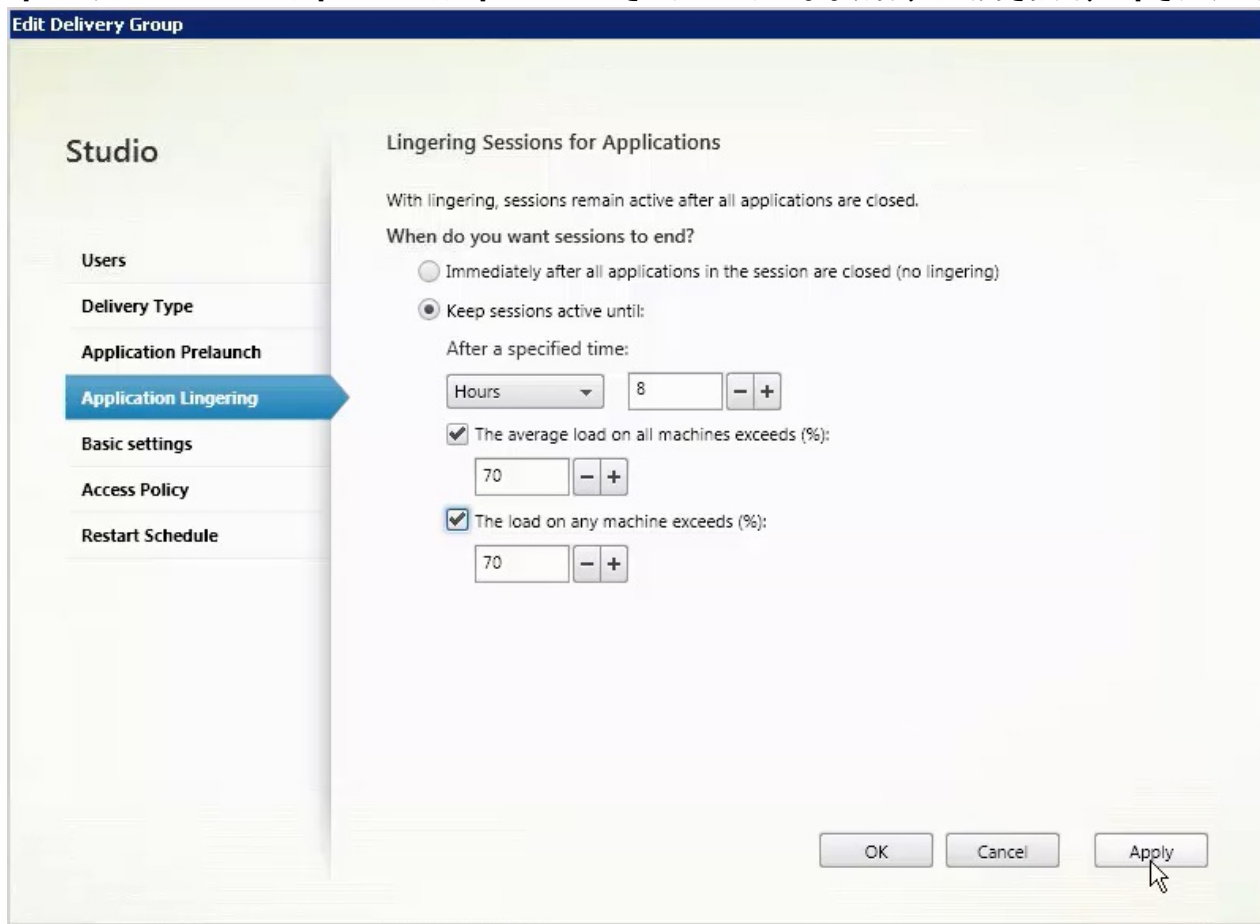
—事前起動セッションおよび残留セッションを保持する時間  
」を参照してください。

- この時間が経過したときにセッションを終了する。セッションを自動的に終了するまでの時間を1~99日間、1~2,376時間、または1~142,560分で指定します。
- デリバリーグループ内のすべてのマシンの平均負荷が上限値（1~99%）を超えたときに終了する。
- デリバリーグループ内のいずれかのマシンの負荷が上限値（1~99%）を超えたときに終了する。

事前起動セッションは、ユーザーがいずれかのアプリケーションを起動したとき、指定した時間が経過したとき、または指定した負荷のしきい値を超えたときのいずれかの状態が発生するまで保持されます。

セッション残留を有効にするには、次の手順に従います。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. デリバリーグループを選択して、[操作] ペインの [デリバリーグループの編集] を選択します。
3. [アプリケーションの残留] ページで、[セッションをアクティブのまま保持する期間を指定する] をクリックします。



4. ユーザーが別のアプリケーションを起動しない場合、残留セッションを保持する時間は複数の設定によって決定されます。これらの設定について詳しくは、次の「

—事前起動セッションおよび残留セッションを保持する時間  
」を参照してください。

- この時間が経過したときにセッションを終了する。セッションを自動的に終了するまでの時間を1~99日間、1~2,376時間、または1~142,560分で指定します。

- デリバリーグループ内のすべてのマシンの平均負荷が上限値（1～99%）を超えたときに終了する。
- デリバリーグループ内のいずれかのマシンの負荷が上限値（1～99%）を超えたときに終了する。

残留セッションは、ユーザーがいずれかのアプリケーションを起動したとき、指定した時間が経過したとき、または指定した負荷のしきい値を超えたときのいずれかの状態が発生するまで保持されます。

**事前起動セッションおよび残留セッションを保持する時間** - ユーザーがアプリケーションを起動しない場合に、使用されないセッションをどのくらい保持するかを指定するには、タイムアウトおよびサーバー負荷のしきい値を構成します。タイムアウトおよびサーバー負荷のしきい値をすべて設定することができます。この場合、最初に発生したイベントによって事前起動セッションや残留セッションが終了します。

- **タイムアウト** — 使用されない事前起動セッションや残留セッションを保持する日数、時間数、または分数を指定できます。この値が短すぎるとセッションがすぐに終了してしまい、ユーザーがアプリケーションにすばやくアクセスできるというメリットが活かされません。また、タイムアウト値が長すぎると、サーバーのリソースが足りなくなり、ユーザーの接続要求が拒否される場合があります。

このタイムアウトをStudioで無効にすることはできませんが、SDK（New/Set-BrokerSessionPreLaunchコマンドレット）で無効にできます。タイムアウトを無効にすると、Studioにタイムアウトが表示されなくなります。

- **しきい値** — サーバーの負荷が高くなったときに事前起動セッションや残留セッションを自動的に終了することができます。これにより、サーバーの負荷が低い間は可能な限りセッションが保持されます。新しいユーザーセッション用のリソースが必要になったときに事前起動セッションや残留セッションが自動的に終了するため、これらのセッションが原因で接続が拒否されることはありません。

デリバリーグループ内の全サーバーの平均負荷パーセンテージと、デリバリーグループ内のいずれかのサーバーの最大負荷パーセンテージの2つのしきい値を構成できます。サーバーの負荷がいずれかのしきい値を超えると、最も長い時間保持された事前起動セッションまたは残留セッションが終了します。その後、負荷がしきい値を下回るまで、分間隔で1つずつセッションが終了します。しきい値を超えている間は、新たな事前起動セッションは開始されません。

Controllerに登録されていないVDAが動作するサーバーやメンテナンスモードのサーバーは、負荷限界状態として認識されません。サーバーで計画外の停止状態が発生した場合、事前起動セッションや残留セッションは自動的に終了してリソースが解放されます。

# XenAppの公開アプリケーションと公開デスクトップ

Jan 04, 2017

XenAppの公開アプリケーションと公開デスクトップは、サーバーOSマシンを使用してユーザーに配信します。

次の表は、これらの配信方法の使用例、目的、対象ユーザー、および注意事項について示しています。

|                       |                                                                                                                                                                                                                                                               |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>使用例</b>            | <b>使用目的</b><br>安価なサーバーベースのアプリケーション配信により、最小限のコストでアプリケーションを多くのユーザーに配信し、しかも高度なセキュリティと良好なユーザーエクスペリエンスを提供する。<br><b>ユーザーの種類</b><br>明確に定義されたタスクだけを実行し、個人用設定やオフラインアクセスが不要なユーザー。たとえば、コールセンターのオペレーター、販売員、ワークステーションを共有する作業員など。<br><b>アプリケーションの種類</b><br>あらゆる種類のアプリケーション。 |
| <b>特長と注意事項</b>        | <b>長所</b><br>データセンター内で簡単に管理できるスケーラブルなソリューション。<br>最もコスト効率に優れたアプリケーション配信ソリューション。<br>ホスト上のアプリケーションを一元管理でき、ユーザーはアプリケーションを変更できません。また、安全で信頼性が高く一貫したユーザーエクスペリエンスが提供されます。<br><b>注意事項</b><br>アプリケーションにアクセスするユーザーは常にオンライン状態である必要があります。                                  |
| <b>ユーザーエクスペリエンス</b>   | ユーザーは、StoreFront、 [スタート] メニュー、または特定のURLからアプリケーションにアクセスします。<br>アプリケーションはユーザーデバイス上に仮想的に配信され、シームレスかつ高品位に表示されます。<br>プロファイル設定によっては、ユーザーによる変更内容がアプリケーションセッションの終了時に保存されません。それ以外の場合、変更は削除されます。                                                                        |
| <b>プロセス、ホスト、および配信</b> | <b>プロセス</b><br>アプリケーションのプロセスはユーザーデバイスではなくホストマシン上で実行されます。<br>物理マシンまたは仮想マシンでアプリケーションをホストできます。<br><b>ホスト</b>                                                                                                                                                     |

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 信            | <p>アプリケーションおよびデスクトップはサーバーOSマシン上にインストールされます。</p> <p>マシンは、マシンカタログを作成することで使用可能になります。</p> <p><b>配信</b></p> <p>マシンカタログ内のマシンは、同じアプリケーションのセットをユーザーグループに配信するデリバリーグループとしてグループ化されます。</p> <p>サーバーOSマシンでは、以下のデリバリーグループを作成できます。</p> <ul style="list-style-type: none"> <li>• デスクトップとアプリケーションの両方をホストする「デスクトップおよびアプリケーション」のデリバリーグループ</li> <li>• アプリケーションだけをホストする「アプリケーション」のデリバリーグループ</li> </ul>                                                                                                                                                                                          |
| セッション管理と割り当て | <p><b>セッション</b></p> <p>サーバーOSマシンは、単一マシン上で複数のセッションを実行して、同時に接続する複数のユーザーに複数のアプリケーションとデスクトップを配信します。各ユーザーは、単一セッション内ですべてのアプリケーションを実行します。</p> <p>たとえば、ユーザーがログオンしてアプリケーションを要求すると、そのマシン上で1つのセッションがホストされ、ほかのユーザーはそのセッションを使用できません。2人目のユーザーが同じマシンにログオンしてアプリケーションを要求すると、2つ目のセッションがホストされ、ほかのユーザーが使用できないセッションが2つになります。これらの2人のユーザーがさらにアプリケーションを要求しても、既存のセッションを使用できるため追加のセッションはホストされません。さらに別の2人のユーザーがログオンしてデスクトップを要求すると、このマシンでは4つのセッションが4人のユーザー用にホストされます。</p> <p><b>ランダムなマシン割り当て</b></p> <p>ユーザーが割り当てられるデリバリーグループ内で、最も負荷が軽いサーバー上のマシンが選択されます。ユーザーのログオン時に、アプリケーション配信用のマシンがランダムに割り当てられます。</p> |

XenAppの公開アプリケーションを配信するには、次の手順に従います。

1. サポートされるWindowsサーバーOSのマスターイメージ上に、配信するアプリケーションをインストールします。
2. このマスターイメージのマシンカタログを作成するか、既存のマシンカタログをこのマスターイメージで更新します。
3. アプリケーションをユーザーに配信するためのデリバリーグループを作成します。
4. インストール済みアプリケーションの一覧で、配信するアプリケーションを選択します。

XenAppの公開デスクトップを配信するには、次の手順に従います。

1. サポートされるWindowsサーバーOSのマスターイメージ上にアプリケーションをインストールします。
2. このマスターイメージのマシンカタログを作成するか、既存のマシンカタログをこのマスターイメージで更新します。
3. デスクトップをユーザーに配信するためのデリバリーグループを作成します。

# VM Hosted Apps

Jan 04, 2017

VM Hosted Appとは、デスクトップOSマシンを使用してユーザーに配信するアプリケーションを指します。

次の表は、この配信方法の使用例、目的、対象ユーザー、および注意事項について示しています。

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>使用例</b>     | <p><b>使用目的</b><br/>セキュアなクライアントベースのアプリケーション配信により、シームレスかつ高品位に表示されるアプリケーションをユーザーに配信し、管理を一元化して単一ホストサーバー（またはハイパーバイザー）で多くのユーザーをサポートする。</p> <p><b>ユーザーの種類</b><br/>内部または外部契約社員、サードパーティの協力者、臨時社員など。<br/>ホスト上のアプリケーションへのオフラインアクセスは不要。</p> <p><b>アプリケーションの種類</b><br/>ほかのアプリケーションと共存できないアプリケーションや、オペレーティングシステムと一緒に動作するMicrosoft .NET Frameworkなどのアプリケーション。これらのアプリケーションは、仮想マシン上でのホストに適しています。<br/>Windows XPやWindows Vistaなど以前のバージョンのオペレーティングシステムや、32ビットや16ビットの古いアーキテクチャで動作するアプリケーション。各アプリケーションを専用の仮想マシン上で分離して、1つの仮想マシンに障害が発生してもほかのユーザーに影響しないようにします。</p>                                         |
| <b>特長と注意事項</b> | <p><b>長所</b><br/>マスターイメージ上のアプリケーションおよびデスクトップは、データセンター内のマシン上でセキュアに管理、ホスト、および実行されます。また、最もコスト効率に優れたアプリケーション配信ソリューションでもあります。</p> <ul style="list-style-type: none"><li>• ユーザーがログオンすると、同じアプリケーションをホストするデリバリーグループ内のマシンにランダムに割り当てられます。</li><li>• 管理者は、ユーザーがログオンするたびに同じマシンが割り当てられるように構成することもできます。このようにマシンをユーザーに静的に割り当てると、ユーザーが仮想マシンにアプリケーションをインストールしたり独自に管理したりできるようになります。</li></ul> <p><b>注意事項</b><br/>デスクトップOSマシンでは、複数のセッションを実行できません。このため、ユーザーがログオンするとデリバリーグループ内の1つのマシンが消費され、オフライン状態ではアプリケーションにアクセスできなくなります。<br/>この方法では、アプリケーションの処理のためのサーバーリソースと、ユーザーのPersonal vDisk用のストレージ容量が多くなります。</p> |

|                            |                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ユーザー<br>エクスペ<br>リエンス       | サーバーOSマシン上でホストされる共有アプリケーションと同様のシームレスなユーザーエクスペリエンスが提供されます。                                                                                                                                                                                                                                                                                                                                 |
| プロセ<br>ス、ホス<br>ト、およ<br>び配信 | <p><b>プロセス</b></p> <p>仮想デスクトップOSマシンが使用されることを除き、サーバーOSマシンの場合と同様です。</p> <p><b>ホスト</b></p> <p>仮想デスクトップOSマシンが使用されることを除き、サーバーOSマシンの場合と同様です。</p> <p><b>配信</b></p> <p>デスクトップOSマシンが「デスクトップのデリバリーグループ」内にのみに追加可能であることを除き、サーバーOSマシンの場合と同様です。</p>                                                                                                                                                     |
| セッショ<br>ン管理と<br>割り当て       | <p><b>セッション</b></p> <p>1つのデスクトップOSマシンで実行できるデスクトップセッションは1つのみです。アプリケーションにのみアクセスする場合は、各アプリケーションが個別のセッションと見なされるため、単一ユーザーが複数のアプリケーションを使用できます。</p> <p><b>ランダムおよび静的なマシン割り当て</b></p> <p>デリバリーグループでは、以下のマシンをユーザーに割り当てることができます。</p> <ul style="list-style-type: none"> <li>• 静的に割り当てられるマシン。ユーザーは、毎回同じマシンにログオンします。</li> <li>• ランダムに割り当てられるマシン。ユーザーは、セッションが使用可能かどうかにより任意に選択されるマシンにログオンします。</li> </ul> |

VM Hosted Appを配信するには、次の手順に従います。

1. サポートされるWindowsデスクトップOSのマスターイメージ上に、配信するアプリケーションをインストールします。
2. このマスターイメージのマシンカタログを作成するか、既存のマシンカタログをこのマスターイメージで更新します。  
マシンカタログのデスクトップエクスペリエンスを定義するときに、ユーザーがログオンするたびに新しい仮想マシンに  
続するのか同じ仮想マシンに接続するのかを指定します。
3. アプリケーションをユーザーに配信するためのデリバリーグループを作成します。
4. インストール済みアプリケーションの一覧で、配信するアプリケーションを選択します。

# VDIデスクトップ

Jan 04, 2017

デスクトップOSマシンを使用してユーザーにVDIデスクトップを配信します。

VDIデスクトップは、仮想マシン上でホストされ、各ユーザーにデスクトップオペレーティングシステムを提供します。

VDIデスクトップでは、XenAppの公開デスクトップよりも多くのリソースが必要になります。ただし、サーバーオペレーティングシステムをサポートしないアプリケーションをインストールできる点が公開デスクトップと異なります。また、VDIデスクトップの種類によっては、特定のユーザーにデスクトップを割り当てて、個々のユーザーによるデスクトップのパーソナル設定を許可できます。

VDIデスクトップのマシナカタログを作成するときは、以下のいずれかの種類のデスクトップを作成します。

- ランダムな非永続デスクトップ。「プール型のVDIデスクトップ」とも呼ばれます。この種類のデスクトップでは、ユーザーがログオンするたびに、単一のマスターイメージに基づいたデスクトッププールからデスクトップが動的に選択されます。デスクトップに対するユーザーの変更内容は、マシンの再起動時に破棄されます。
- 静的な非永続デスクトップ。この種類のデスクトップでは、ユーザーの初回ログオン時に、単一のマスターイメージに基づいたデスクトッププールからデスクトップが割り当てられます。以降のログオンでは、同じデスクトップに接続されます。デスクトップに対するユーザーの変更内容は、マシンの再起動時に破棄されます。
- 静的な永続デスクトップ。Personal vDiskを使用するVDIデスクトップです。ほかの種類のVDIデスクトップとは異なり、ユーザーによるデスクトップのパーソナル設定が許可されます。この種類のデスクトップでは、ユーザーの初回ログオン時に、単一のマスターイメージに基づいたデスクトッププールからデスクトップが割り当てられます。以降のログオンでは、同じデスクトップに接続されます。デスクトップに対するユーザーの変更内容はPersonal vDisk内に保持されるため、マシンを再起動しても破棄されません。

VDIデスクトップは、以下の手順で配信します。

1. WindowsデスクトップOSを実行するマスターイメージを作成します。
2. このマスターイメージのマシナカタログを作成するか、既存のマシナカタログをこのマスターイメージで更新します。マシナカタログのデスクトップエクスペリエンスを定義するときに、ユーザーがログオンするたびに新しい仮想マシンに接続するのか同じ仮想マシンに接続するのかを指定し、さらにデスクトップに対する変更内容の保持方法を指定します。
3. デスクトップをユーザーに配信するためのデリバリーグループを作成します。

# リモートPCアクセス

Jan 04, 2017

リモートPCアクセス機能により、社内にある自分の物理的なWindows PCにどこからでもリモートアクセスできるようになります。社内PC上にインストールするVirtual Delivery Agent (VDA) により、そのPCがDelivery Controllerに登録され、PCとクライアントデバイス間のHDX接続が管理されます。リモートPCアクセス機能ではセルフサービス型の構成がサポートされており、管理者がホワイトリスト（ユーザーによるアクセスを許可するマシンの一覧）を作成しておくことで、各ユーザーが管理者による介入なく自分の社内PCをサイトに追加できます。クライアントデバイス上で動作するCitrix Receiverにより、リモートPCアクセスセッションから社内のPC上のアプリケーションやデータへのアクセスが可能になります。

ユーザーには、任意の数の物理PC、または物理PCと仮想デスクトップの組み合わせを含む、複数のデスクトップを割り当てることができます。

注：リモートPCのスリープモードおよび休止状態モードはサポートされていません。リモートPCアクセスセッションではXenDesktopライセンスが使用され、ほかのXenDesktopセッションと同じ方法でライセンスが消費されます。

## Active Directoryに関する考慮事項

- リモートPCアクセスの展開サイトを構成する前に、Active Directory組織単位 (OU) とセキュリティグループをセットアップして、Active Directoryユーザーアカウントを作成してください。これらのアカウントは、リモートPCアクセス用のデリバリーグループでユーザーを指定するときに使用します。
- マシンカタログにマシンを追加した後でActive Directoryを変更しても、リモートPCアクセス機能には反映されません。必要な場合は、マシンを手動で別のマシンカタログに再割り当てできます。
- Active DirectoryのOUエントリを移動または削除すると、リモートPCアクセスに使用されるOUエントリで整合性の問題が発生して、VDAとマシンカタログまたはデリバリーグループとの関連付けが解除されることがあります。

## マシンカタログおよびデリバリーグループに関する考慮事項

- リモートPCアクセスでは、1つのマシンを複数のマシンカタログやデリバリーグループに関連付けることはできません。
- リモートPCアクセスのマシンカタログに複数のマシンを追加できます。
- マシンカタログ用のマシンアカウントを選択するときは、ほかのマシンカタログ内のマシンとの競合を避けるため、最も低いレベルのOUを選択します。たとえば、Bank/officers/tellersのOUではtellersを選択します。
- リモートPCアクセスのマシンカタログのすべてのマシンを、1つまたは複数のデリバリーグループで割り当てることができます。たとえば、2つのユーザーグループに異なるポリシー設定を適用するには、それらのユーザーを異なるデリバリーグループに追加して、各デリバリーグループ用のHDXポリシーフィルターを構成します。
- IT管理の責任分担がユーザーの地理的な場所、部署、またはそのほかのカテゴリによって細分化されている場合は、マシンやユーザーをカテゴリごとにグループ化して、委任管理を構成します。各管理者には、そのカテゴリ内のマシンカタログに対応するデリバリーグループの両方に対する管理権限が正しく割り当てられていることを確認してください。
- 社内PCでWindows XPが動作している場合は、そのPC用に専用のマシンカタログおよびデリバリーグループを作成します。この場合、Studioでマシンカタログのマシンアカウントを選択するときに、一部のマシンでWindows XPが動作していることを示すチェックボックスをオンにします。

## 展開に関する考慮事項

- リモートPCアクセス展開を作成した後で、従来のVirtual Desktop Infrastructure (VDI) デスクトップまたはアプリケーションを追加することができます。また、リモートPCアクセス展開を既存のVDI展開に追加することもできます。
- 社内のPC上にVDAをインストールするときに、Windowsリモートアシスタンス機能を有効にすることを検討してください。これにより、ヘルプデスク担当者がDirectorからWindowsリモートアシスタンスを使用して、ユーザーのセッションを表示したり操作したりできるようになります。
- 各社内PCへのVDAのインストール方法を検討します。Active DirectoryスクリプトやMicrosoft System Center Configuration Managerなど、電子的なソフトウェアディストリビューションツールを使用することをお勧めします。インストールメディアには、Active Directoryスクリプトのサンプルが収録されています。



- セキュアブート機能は現在サポートされていません。ワークステーションVDAを展開しようとする場合は、セキュアブートを無効にします。
  - 各社内PCは、有線ネットワーク接続によりドメインに参加している必要があります。
  - 社内PCでのWindows 7 Aeroがサポートされていますが、この機能を使用しなければならないということではありません。
  - キーボードやマウスは、社内のPCやラップトップに直接接続してください。電源がオフになることのあるモニターなどのデバイス経由で接続しないでください。（キーボードやマウスをモニターなどのデバイス経由で接続する必要がある場合は、そのデバイスの電源がオフにならないようにする必要があります）
  - スマートカードを使用する場合は、「[スマートカード](#)」を参照してください。
  - リモートPCアクセスはほとんどのラップトップコンピューターでも使用できます。いつでもアクセスできるように、ラップトップコンピューターの省電力設定を、デスクトップPCと同様に構成する必要があります。次に例を示します。
    - 休止機能を無効にする。
    - スリープ機能を無効にする。
    - カバーを閉じた場合の動作を [何もしない] に設定する。
    - [電源ボタンを押したときの動作] を [シャットダウン] に設定する。
    - ビデオカードの省電力設定を無効にする。
    - インターフェイスカードの省電力設定を無効にする。
    - バッテリーの節電機能を無効にする。
- 以下のデバイスは、リモートPCアクセスでサポートされません。

- ラップトップのドッキングおよびドッキング解除。
- KVMスイッチ、またはセッションを切断する可能性のあるその他のコンポーネント。
- ハイブリッドPC（オールインワンおよびNVIDIA OptimusラップトップおよびPCを含む）およびSurface Pro/Books。
- 社内PCにリモートアクセスする各クライアントデバイスにCitrix Receiverをインストールします。
- 同じ社内PCにリモートアクセスする複数のユーザーには、Receiverに同じアイコンが表示されます。いずれかのユーザーがそのPCにリモートからログオンすると、ほかのユーザーにはそれを使用できないことを示すアイコンが表示されます。
- デフォルトでは、ローカルユーザーがそのマシンでCtrl+Alt+Delキーを押してセッションを開始すると、リモートユーザーのセッションは自動的に切断されます。自動的に切断されないようにするには、社内PCに次のレジストリエントリを追加してから、マシンを再起動します。

注意：レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

HKLM\SOFTWARE\Citrix\PortICA\RemotePC "SasNotification"=dword:00000001

HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\PortICA\RemotePCで、この機能の内容をよりカスタマイズするには

- RpcaMode (dword)
- RpcaTimeout (dword)

RpcaMode :

- 1 - 指定のタイムアウト期間にMessaging UIへ応答しない場合、リモートユーザーは常に獲得します。
- 2 - ローカルユーザーが常に獲得します。この設定を指定しない場合、リモートユーザーがデフォルトで獲得します。

RpcaTimeout :

どの種類のモードが適用されるか自動的に決定される前に、2つ目の数がユーザーに提供されます。この設定が指定されない場合、デフォルト値は30秒です。この最小値は、30秒とする必要があります。変更を有効にするために、ユーザーはマ

シンを再起動する必要があります。

ユーザーがコンソールへ強制的にアクセスできるようにするには、ローカルユーザーがCtrl+Alt+Delキーを2回、10秒の間、押し、リモートセッション上でローカルの制御を実行できるようにして、切断イベントを強制的に実行します。

このレジストリ設定により、リモートユーザーが社内PCのセッションを使用しているときにローカルユーザーがCtrl+Alt+Delキーを押してそのPCにログオンすると、ローカルユーザーの接続を許可するかどうかを確認するメッセージがリモートユーザーに表示されます。接続を許可すると、リモートユーザーのセッションは切断されます。

以下のXenDesktop機能は、リモートPCアクセスでサポートされません。

- マスターイメージおよび仮想マシンの作成
- ホストされるアプリケーションの配信
- Personal vDisk
- クライアントフォルダーのリダイレクト

リモートPCアクセスではWake on LANがサポートされ、物理PCをリモートから起動できます。この機能により、ユーザーが退社時にPCの電源をオフにできるようになるため、消費電力を節約できます。また、停電や気象条件などにより電源がオフになったPCにもリモートアクセスできるようになります。

XenDesktop 7.6 Feature Pack 3により、Wake on LAN SDKを実行できます。これにより、Wake on LANソリューションを有効にして、System Center 2012 R2を必要とすることなくコネクタを作成できます。詳しくは、<http://support.citrix.com/article/CTX202272>を参照してください。

リモートPCアクセスのWake on LAN機能は、以下のデバイスでサポートされます。

- Intel社のActive Management Technology (AMT) をサポートするPC。
- BIOSでWake on LANオプションが有効になっているPC。

管理者は、Microsoft System Center Configuration Manager (ConfigMgr) 2012のWake on LAN機能を構成する必要があります。ConfigMgrでは、PCに対してAMTパワーコマンドを実行できます。また、ウェイクアッププロキシやマジックパケットがサポートされます。さらに、StudioでリモートPCアクセス展開を作成するとき（またはリモートPCアクセス用の電源管理接続を追加するとき）に、電源管理機能を有効にして、ConfigMgrのアクセス情報を指定します。

さらに、以下を確認します。

- セキュリティおよび信頼性の高いAMTパワーコマンドの使用が推奨されますが、AMTを使用せずにConfigMgrのウェイクアッププロキシやマジックパケットを使用する方法もあります。
- AMTに対応したマシンでは、StudioやDirectorからそのマシンを強制シャットダウンまたは強制再起動することもできます。また、StoreFrontおよびReceiverでは再起動操作も使用できます。

詳しくは、「[Configuration ManagerとリモートPCアクセスのWake on LAN](#)」および「[ユーザーへのリモートPCアクセスの提供](#)」を参照してください。

# ユーザーへのリモートPCアクセスの提供

Jan 04, 2017

ユーザーがリモートPCデスクトップを使用すると、社内の自分のPC上のリソースにリモートから安全にアクセスできます。このとき、Citrix HDXテクノロジーによる優れたユーザーエクスペリエンスが提供されます。

注：リモートPCアクセスを使用するには、XenDesktopライセンスが必要です。

1. リモートPCアクセスの電源管理機能（リモートPCアクセスのWake on LAN機能）を使用するには、StudioでリモートPCアクセス展開を作成する前に、接続先のPCとMicrosoft System Center Configuration Manager (ConfigMgr) 2012での構成を完了しておく必要があります。詳しくは、「[Configuration ManagerとリモートPCアクセスのWake on LAN](#)」を参照してください。
2. 管理者は、最初のリモートPCアクセス展開の作成時に、デフォルトのリモートPCアクセスマシンカタログのマシンの電源管理機能を有効または無効にできます。電源管理を有効にする場合は、ConfigMgr接続の情報を指定します。さらに、ユーザーおよびマシンのアカウントを指定します。詳しくは、「[サイトの作成](#)」を参照してください。リモートPCアクセス展開を作成した後も、VDI展開を作成できます。  
リモートPCアクセス展開を作成すると、「  
—リモートPCアクセスマシン  
」という名前のデフォルトマシンカタログと、「  
—リモートPCアクセスデスクトップ  
」という名前のデフォルトデリバリーグループが作成されます。
3. リモートPCアクセス用のマシンカタログを追加する場合は、以下の操作を行います。
  - オペレーティングシステム：[リモートPCアクセス]を選択し、電源管理接続を選択します。電源管理機能を無効にすることもできます。構成済みの電源管理接続がない場合は、マシンカタログの作成ウィザードを完了した後で接続を作成し（接続の種類は[Microsoft Configuration Manager Wake on LAN]）、マシンカタログを編集してその接続を指定します。
  - マシンアカウント：一覧からマシンアカウントまたは組織単位（OU）を選択するか、マシンアカウントおよびOUを追加します。
4. ローカルおよびリモートアクセスで使用する社内PCにVDAをインストールします。一般的にはパッケージ管理ソフトウェアを使用してVDAを自動的に展開します。ただし、検証用または小規模な展開の場合は、各社内PC上にVDAを手動でインストールできます。  
VDAをインストールした後で、その社内PCのコンソールセッションに（ローカルまたはRDP経由で）最初にログオンしたドメインユーザーにそのリモートPCデスクトップが自動的に割り当てられます。そのコンソールセッションにさらにほかのドメインユーザーがログオンすると、それらのユーザーもデスクトップユーザーの一覧に追加されます。ただし、これらのユーザーには管理者が構成した制限事項が適用されます。  
注：XenApp/XenDesktop環境外でRDP接続を使用するには、ユーザーまたはグループをDirect Access Usersグループに追加する必要があります。
5. 社内PCへのリモート接続で使用するすべてのユーザーデバイスにCitrix Receiverをダウンロードおよびインストールするようユーザーに指示します。Citrix Receiverは<http://www.citrix.com>から、または使用するモバイルデバイスのアプリケーション配布システム（アプリケーションストア）から入手できます。

電源管理接続を編集して、詳細設定を変更できます。以下の機能を有効にできます。

- ConfigMgrのウェイクアッププロキシ。
- Wake on LAN（マジック）パケット。Wake on LANパケットを有効にする場合は、パケットの転送方法としてサブネット向けのブロードキャストまたはユニキャストを選択できます。

社内PCではAMTパワーコマンド（サポートされる場合）と、有効にした詳細設定が使用されます。AMTパワーコマンドが使用されない場合は、詳細設定が使用されます。

Delivery Controllerは、リモートPCアクセスに関する以下の診断情報をWindowsのアプリケーションイベントログに書き込みます。情報メッセージは調整されません。エラーメッセージは重複メッセージの破棄により調整されます。

- 3300 (情報) — マシンカタログへのマシンの追加
- 3301 (情報) — デリバリーグループへのマシンの追加
- 3302 (情報) — ユーザーへのマシンの割り当て
- 3303 (エラー) — 例外の発生

リモートPCアクセス用の電源管理を有効にすると、サブネット向けのブロードキャストでのマシンの起動に失敗することがあります。この問題は、Controllerとマシンが異なるサブネット上に存在する場合に発生します。AMTがサポートされない場合に異なるサブネット間でサブネット向けのブロードキャストを使用するには、ウェイクアッププロキシまたはユニキャストを使用してください。これらの詳細設定は、電源管理接続のプロパティで有効にできます。

# リモートPCアクセスのデリバリーグループの管理

Jan 04, 2017

リモートPCアクセス用のマシンカタログでユーザーに割り当てられていないマシンは、そのマシンカタログに関連付けられたデリバリーグループに一時的に割り当てられます。この一時的な割り当てにより情報が提供されるため、そのマシンを後で特定のユーザーに割り当てることができます。デリバリーグループとマシンカタログとの関連付けには優先度値があります。

この優先度により、マシンをシステムに登録したりユーザーにマシンを割り当てたりするときのデリバリーグループが決定されます。この値が小さいほど優先度が高くなります。リモートPCアクセスマシンカタログに複数のデリバリーグループ割り当てがある場合、優先度が最も高い割り当てが選択されます。この優先度値はPowerShell SDKを使用して設定できます。

リモートPCアクセス用のマシンカタログの初回作成時に、デリバリーグループが関連付けられます。つまり、このマシンカタログに後から追加したコンピューターアカウントまたは組織単位を、このデリバリーグループに追加することができます。この関連付けは、必要に応じて有効にしたり無効にしたりできます。

1. Studioのナビゲーションペインで [デリバリーグループ] を選択します。
2. リモートPCアクセスのデリバリーグループを選択します。
3. [詳細] セクションで [マシンカタログ] タブを選択し、リモートPCアクセス用のマシンカタログを選択します。
4. 関連付けを追加または復元するには、[デスクトップの追加] を選択します。関連付けを削除するには、[関連付けの削除] を選択します。

# App-V

Jan 04, 2017  
Microsoft Application Virtualization (App-V) を使用すると、アプリケーションをサービスとして展開、更新、およびサポートできます。ユーザーは、自分のデバイスにインストールすることなくこれらのアプリケーションにアクセスできます。App-VおよびMicrosoft User State Virtualization (USV) では、場所やインターネット接続を問わずに、ユーザーにアプリケーションやデータへのアクセスを提供できます。

次の表は、サポートされるバージョンの一覧です。(App-V 4.6 SP2のクライアントはサポートされません)

| App-V         | XenDesktopおよびXenAppのバージョン          |         |
|---------------|------------------------------------|---------|
|               | Delivery Controller                | VDA     |
| 5.0           | XenDesktop 7~現在の<br>XenApp 7.5~現在の | 7.0~現在の |
| 5.0 SP1       | XenDesktop 7~現在の<br>XenApp 7.5~現在の | 7.0~現在の |
| 5.0 SP2       | XenDesktop 7~現在の<br>XenApp 7.5~現在の | 7.1~現在の |
| 5.0 SP3および5.1 | XenDesktop 7.6<br>XenApp 7.6       | 7.6.300 |

サポートされるApp-Vのクライアントは、アプリケーションへのオフラインアクセスをサポートしません。App-Vの統合機能により、アプリケーションでのSMB共有の使用がサポートされます。HTTPプロトコルはサポートされません。

- アプリケーションは、事前構成やオペレーティングシステム設定の変更を行わなくても、シームレスに使用可能になります。App-Vは、以下のコンポーネントで構成されています。
- Management Server (管理サーバー) — App-Vインフラストラクチャを管理したり、App-Vデスクトップクライアントやリモートデスクトップサービスクライアントに仮想アプリケーションを配信したりするための中央管理コンソールです。App-V管理サーバーは、セキュリティ、測定、監視、および管理者によって要求されるデータ収集を認証、要求、および提供します。このサーバーは、Active Directoryといくつかのツールを使用してユーザーとアプリケーションを管理します。
  - Publishing Server (公開サーバー) — App-V Clientに特定ユーザー用のアプリケーションを提供し、ストリーム配信用の仮想アプリケーションパッケージをホストします。これらのパッケージは、管理サーバーから取得されます。
  - クライアント — 公開サーバーから仮想アプリケーションを取得したり、クライアント上のアプリケーションを公開したり、Windowsデバイス上で仮想環境を自動的にセットアップおよび管理したりします。App-VクライアントはVDA上にインストールされ、各ユーザーのプロファイルにおけるレジストリやファイルの変更など、ユーザー固有の仮想アプリケーション設定を格納します。

以下の方法で、サーバーOSおよびデスクトップOSのデリバリーグループからApp-Vアプリケーションを起動できます。

- Citrix Receiverを使用して起動する。
- [スタート]メニューから起動する。
- App-VクライアントおよびCitrix Receiverを使用して起動する。
- 複数のデバイス上のアプリケーションを複数ユーザーが同時に起動する。
- Citrix StoreFrontから起動する。

App-Vアプリケーションのプロパティに対する変更は、そのアプリケーションの起動時に適用されます。たとえば、アプリケーションの表示名やアイコンを変更した場合、ユーザーがそのアプリケーションを起動すると変更内容が表示されます。

「デスクトップおよびアプリケーション」配信のデリバリーグループを「アプリケーション」配信のデリバリーグループに変換しても、App-Vアプリケーションのパフォーマンスは変わりません。

管理者がApp-Vの管理サーバーおよび公開サーバーでApp-Vアプリケーションを管理する、App-Vサーバーベース展開のみがサポートされます。

App-Vアプリケーションを配信するには

1. <http://technet.microsoft.com/en-us/virtualization/hh710199>の説明を参照して、App-Vを展開します。
2. App-V管理サーバー上で、App-Vアプリケーションを公開します。アクセス権やファイルタイプの関連付けなどの機能を構成します。既存のApp-Vの展開環境では、これらの機能は構成済みです。
3. 必要に応じて、App-V公開サーバーを変更します。以下を参照してください。
4. App-VクライアントをVDA上にインストールします。
5. Studioでサイトを構成するときに、App-Vの公開サーバーおよび管理サーバーのURLとポート番号を指定します。これらのサーバーは、デリバリーグループにより自動的に使用されます。
6. App-Vクライアントを、Studioのマシンのカタログで使用するマスターイメージ上にインストールします。App-VクライアントにShareContentStoreModeおよびEnablePackageScriptsなどの設定を成する必要があります(マスターイメージではApp-V公開サーバーを構成する必要はありません。公開サーバーはアプリケーションの起動時に構成されます)。
7. デリバリーグループを作成するときに、App-Vアプリケーションを選択します。

これで、アプリケーションが使用できるようになりました。

サイトを作成した後で、App-Vのサーバー情報を指定または変更できます。Studioのナビゲーションペインで、[構成] の [App-V公開] を選択し、[操作] ペインでエントリを選択します。App-V公開を追加するときは、App-Vの管理サーバーおよび公開サーバーのURLとポート番号を指定します。これらのアドレスを編集または削除することもできます。App-Vアプリケーションを更新した場合、サーバーに関する問題があるかどうか画面に表示され、使用できなくなったアプリケーションのエントリが削除されます。

公開サーバーの設定を変更するには、Controller上でSDKコマンドレットを使用することをお勧めします。

- 公開サーバーの設定を表示するには、Get-CtxAppvServerSetting -AppVPublishingServerと入力します。
- App-Vアプリケーションが正しく起動するようにするには、Set-CtxAppvServerSetting -UserRefreshOnLogon 0と入力します。

Controller上でApp-V公開サーバーの設定を変更するには、次のコマンドレットを実行します。一部のパラメーターはオプションです。

Set-CtxAppvServerSetting -AppVPublishingServer -UserRefreshOnLogon -UserRefreshEnabled -UserRefreshInterval -UserRefreshIntervalUnit -GlobalRefreshOnLogon -GlobalRefreshEnabled -GlobalRefreshInterval

注: GPOポリシーを使って公開サーバー設定を管理していた環境では、上記のコマンドレットでの設定を含み、すべてのApp-V統合設定よりもGPOでの設定が優先されてしまいます。このため、

App-Vアプリケーションの起動に失敗することがあります。この問題を避けるため、すべてのGPOポリシー設定を削除し、SDKを使って同じ設定を構成することを推奨します。

- StudioでApp-Vの管理サーバーと公開サーバーのアドレスを指定するときの接続テストに失敗する場合は、以下を確認してください。
  - App-Vサーバーの電源が入っていること。pingコマンドを送信するか、インターネットインフォメーションサービス (IIS) マネージャーで各App-Vサーバーの状態が開始済みであることを確認します。
  - App-VサーバーのPowerShellリモート処理が有効であること。有効になっていない場合は、<http://technet.microsoft.com/en-us/magazine/ff700227.aspx>の説明に従って有効にします。
  - App-VサーバーがActive Directoryに追加されていること。  
StudioのマシンとApp-Vサーバーが信頼関係のない異なるActive Directoryドメインに属している場合は、Studioマシン上のPowerShellコンソールでwinrm s winrm/Config/client '@(TrustedHosts="`<App-V server FQDN>`")'を実行します。TrustedHostsがGPOで管理されている場合は、「構成設定TrustedHostsはポリシーで制御されているため変更できません。構成設定を変更するには、ポリシーを「未構成」に設定する必要があります」というエラーメッセージが表示されます。このメッセージが表示された場合は、GPOのTrustedHostsポリシー（[管理用テンプレート] > [Windowsコンポーネント] > [Windowsリモート管理 (WinRM)] > [WinRMクライアント]）にApp-Vサーバーの名前を追加します。
  - Studioの管理者がApp-Vサーバーの管理者でもあること。
  - App-Vサーバー上でファイル共有が有効になっていること。Windowsエクスプローラーまたは[ファイル名を指定して実行]ダイアログボックスに「\\<App-V Server FQDN>」と入力します。
  - App-VサーバーにApp-V管理者と同じファイル共有権限が付与されていること。App-Vサーバーで、[ユーザー名およびパスワードの保存]ダイアログボックスに「\\<App-V Server FQDN>」のエントリを追加して、そのApp-Vサーバーの管理者権限を持つユーザーの資格情報を指定します。詳しくは、<http://support.microsoft.com/kb/306541>を参照してください。
- アプリケーションの検出に失敗する場合は、以下の点を確認してください。
  - Studioの管理者がApp-V管理サーバー上の管理者でもあること。
  - App-V管理サーバーが動作していること。インターネットインフォメーションサービス (IIS) マネージャーを開いて、サーバーが正しく動作していることを確認します。
  - App-VサーバーのPowerShellリモート処理が有効であること。これらのサーバーで有効になっていない場合は、<http://technet.microsoft.com/en-us/magazine/ff700227.aspx>の説明に従って有効にします。
  - Studio管理者がアクセスできるように、パッケージに適切なセキュリティ権限があるかどうか。
- App-Vアプリケーションが起動しない場合は、以下の点を確認してください。
  - 公開サーバーが動作していること。インターネットインフォメーションサービス (IIS) マネージャーを開いて、サーバーが正しく動作していることを確認します。
  - App-Vパッケージに適切なセキュリティ権限が設定されており、ユーザーがアクセスできるようになっているかどうか。
  - VDA上で以下を確認します。
    - Tempディレクトリの参照が正しく、そのディレクトリに十分な空き領域があるかどうか。
    - App-V 5.0以降のクライアントソフトウェアがインストールされているかどうか。
    - 管理者権限でGet-AppvClientConfigurationを実行し、EnablePackageScriptsが1に設定されていることを確認します。1に設定されてない場合は、Set-AppvClientConfiguration - EnablePackageScripts Trueを実行します。  
マスターイメージの作成時にこれを実行することを推奨します。これにより、マスターイメージから作成されるすべてのVDAにこの設定が適用されます。
    - レジストリエディター (regedit) でHKEY\_LOCAL\_MACHINE\SOFTWARE\Policies\Citrix\AppVを開きます。AppVServersキーが「+」形式（「http://xmas-demo-appv.blrstm.com+0+0+0+1+1+1+0+1/http://xmas-demo-appv.blrstm.com:8082」など）であることを確認します。
    - CtxAppVCOMAdminアカウントに管理者権限があるかどうか。通常、CtxAppVCOMAdminはVirtual Delivery Agentのインストール時に作成され、そのVDAマシンのローカル管理者グループに追加されます。ただし、Active Directoryポリシーの設定によっては、このアカウントの管理者権限が無効になることがあります。compmgmt.mscを実行し、[ローカルユーザーとグループ]、[ユーザー]の順に選択します。CtxAppVCOMAdminが管理者ではない場合は、グループポリシーを編集するか管理者に連絡して、このユーザーアカウントに管理権限を付与します。
  - App-Vクライアントがインストールされているマスターイメージでは、Microsoft社により提供されるAppVクライアントモジュールが署名されていないため、PowerShell ExecutionPolicyをRemoteSignedに設定する必要があります。このExecutionPolicy設定により、PowerShellで無署名のローカルのスクリプトおよびコマンドレットを実行できるようになります。次のいずれかの方法を使用してExecutionPolicyを設定します。
    - 管理者としてログインし、次のPowerShellコマンドレットSet-ExecutionPolicy RemoteSignedを入力します。
    - グループポリシー設定で、[コンピューターの構成]、[ポリシー]、[管理用テンプレート]、[Windowsコンポーネント]、[Windows PowerShell]、[スクリプトの実行を有効にする]の順に選択します。
  - 公開サーバーの動作を確認します。
    - Get-AppvPublishingServer \*を実行して、公開サーバーの一覧を表示します。
    - UserRefreshonLogonがFalseに設定されていることを確認します。Falseに設定されていない場合、最初のApp-Vアプリケーションの起動に失敗します。
    - 管理者権限でSet-AppvPublishingServerを実行して、UserRefreshonLogonをFalseに設定します。これらの手順により問題を解決できない場合、ログを有効にして調査する必要があります。

Studioのログを有効にするには、次の手順に従います。

- C:\CtxAppVLogsフォルダーを作成します。
- C:\ProgramFiles\Citrix\StudioAppVInegration\SnapIn\Citrix.Appv.Admin.V1に移動し、管理者としてメモ帳などのテキストエディターでCtxAppVCommon.dll.configを開きます。次の行のコメントを解除します。

VDAのログを有効にするには、次の手順に従います。

- C:\CtxAppVLogsフォルダーを作成します。
- C:\ProgramFiles\Citrix\Virtual Desktop Agentに移動し、管理者としてメモ帳などのテキストエディターでCtxAppVCommon.dll.configを開きます。次の行のコメントを解除します。
- 次の行のコメントを解除して、値フィールドに1を設定します。

構成に関連するすべてのログは、C:\CtxAppVLogsに生成されます。アプリケーション起動ログは、次の場所に生成されます。

- XenDesktop 7.1以降およびXenApp 7.5以降 — %LOCALAPPDATA%\Citrix\CtxAppVLog
- XenDesktop 7.0 — %LocalAppData%\temp\CtxAppVLogs

%LOCALAPPDATA%は、ログオンしたユーザーのローカルフォルダーです。アプリケーション起動に失敗したユーザーのローカルフォルダーでログを確認する必要があります。

- 管理者として、Brokerサービスを再開するか、またはVDAマシンを再起動してログ記録を開始します。

# ローカルアプリケーションアクセスとURLリダイレクト

Jan 04, 2017

ローカルアプリケーションアクセスを有効にすると、ローカルにインストールされているWindowsアプリケーションが仮想デスクトップ環境にシームレスに統合されます。ローカルアプリケーションアクセスにより、以下の操作が可能になります。

- ラップトップやPCなどの物理コンピューター上にローカルにインストールされたアプリケーションに仮想デスクトップからアクセスする。
- フレキシブルなアプリケーション配信ソリューションをユーザーに提供する。仮想化できないアプリケーションやIT担当者が管理しないアプリケーションをユーザーのローカルにインストールして、仮想デスクトップ上にインストールされたアプリケーションのように使用できます。
- 仮想デスクトップ以外のホスト上で提供される公開アプリケーションのダブルホップによる遅延を解消するために、そのアプリケーションのショートカットをユーザーのWindowsデバイス上に配置する。
- 次のようなアプリケーションを使用する。
  - GoToMeetingなどのビデオ会議ソフトウェア。
  - 仮想化されていない特殊なアプリケーション。
  - DVDバーナーやTVチューナーなど、ユーザーデバイスとサーバー間で大量のデータ転送が発生するアプリケーションや周辺機器。

XenAppおよびXenDesktopでローカルアプリケーションアクセスが有効な場合、URLリダイレクトにより、ホストされるデスクトップセッションからローカルアクセスアプリケーションを起動できます。URLリダイレクトでは、複数のURLアドレスでアプリケーションを起動できます。デスクトップセッションで、Webブラウザ内に埋め込まれたリンクをクリックすると、WebブラウザのURLブラックリストに基づいてローカルのWebブラウザが起動します。ブラックリストにないURLをクリックすると、デスクトップセッション内のWebブラウザが再度使用されます。

URLリダイレクトはデスクトップセッションでのみ機能し、アプリケーションセッションでは機能しません。アプリケーションセッションで使用できるリダイレクト機能は、サーバーFTAの1つである「ホストからクライアントへのコンテンツのリダイレクト」のみです。このFTAでは、http、https、rtsp、mmsなど、特定のプロトコルがクライアント側に転送されます。たとえば、httpの埋め込みリンクを開くときに、クライアント側のアプリケーションが使用されます。特定のURLのリダイレクトを有効にしたり無効にしたりすることはできません。

ローカルアプリケーションアクセスを有効にすると、ローカルで実行されるアプリケーション、ホストされるアプリケーション、またはデスクトップ上のショートカットからアクセスされたURLを、以下のいずれかの方法でリダイレクトできます。

- ユーザーのコンピューターから、ホストされているデスクトップへ
- XenApp/XenDesktopサーバーから、ユーザーのコンピューターへ
- 起動された環境内で処理（リダイレクトなし）

特定のWebサイトでのリダイレクト方法を指定するには、Virtual Delivery Agent上のURLホワイトリストおよびURLブラックリストを構成します。これらのリストでは、URLリダイレクトのポリシー設定を指定する複数行文字列値を設定します。詳しくは、「ローカルアプリケーションアクセスのポリシー設定」を参照してください。

すべてのURLをVDA側のWebブラウザで開くこともできますが、以下のURLについてはエンドポイント上のWebブラウザで開くためのポリシーを構成できます。

- ジオ/ロケール情報 — ユーザーの現在位置の情報に基づいて適切なページを自動的に表示するmsn.comやnews.google.comなどのWebサイト。たとえば、イギリスにあるデータセンターで提供されるVDAにインドのクライアントから接続する場合、in.msn.comではなくuk.msn.comが表示されます。
- マルチメディアコンテンツ — メディアリッチなWebサイト。クライアント側で処理されるように設定すると、ユーザー



クスペリエンスが向上し、狭帯域幅接続での使用帯域幅や処理能力が改善されます。Flashリダイレクト機能を使用することもできますが、URLリダイレクトを使用するとSilverlightなどほかの種類のメディアにも対応できます。これにより、環のセキュリティも向上します。つまり、管理者により許可されたURLだけがクライアント側で処理され、ほかのURLはすべてVDA側で処理されます。

URLリダイレクトに加えて、ファイルタイプの関連付け (FTA : File Type Association) によるリダイレクトも使用できます。この機能では、セッションで特定のファイルを開くときにローカルのアプリケーションが使用されます。ローカルアプリケーションでファイルを開くには、そのファイルにアクセスできる必要があります。つまり、ローカルアプリケーションで開くことができるのは、ネットワーク共有上またはクライアントドライブ上にあるファイル (クライアント側ドライブのマッピング機能) のみです。たとえば、PDFファイルを開く場合、ローカルにインストールされているPDFリーダーでファイルが表示されます。ローカルアプリケーションはファイルに直接アクセスできるため、ファイルを開くときにICAによるネットワーク転送は発生しません。

ローカルアプリケーションアクセスはVDA for Windows Server OSおよびVDA for Windows Desktop OSでサポートされるオペレーティングシステムでサポートされ、Citrix Receiver for Windows Version 4.1 (以降) が必要です。次のWebブラウザがサポートされています。

- Internet Explorer 8、9、10、および11
- Firefox 3.5 ~ 21.0
- Chrome 10

ローカルアプリケーションアクセスやURLリダイレクトを使用するときは、以下の考慮事項および制限事項について確認してください。

- ローカルアプリケーションアクセスは全画面モード用に設計されています。このため、以下の制限事項があります。
  - ローカルアプリケーションアクセスをウィンドウ表示モードの仮想デスクトップで使用するなど、単一の仮想デスクトップをすべてのモニター上で表示しない場合、ユーザーエクスペリエンスに混乱が生じます。
  - マルチモニター環境でアプリケーションの表示を1つのモニターで最大化すると、それ以降のアプリケーションはほかのモニターに表示されず、すべてのアプリケーションがそのモニター上に表示されます。
  - この機能は、単一VDAでの使用を想定して設計されています。複数の同時接続VDAを対象とするものではありません。
- 一部のアプリケーションでは、以下の予期されない問題が発生する場合があります。
  - ユーザーが、仮想デスクトップのCドライブとローカルのCドライブを混同する場合があります。
  - 仮想デスクトップで使用できるプリンターは、ローカルアプリケーションでは使用できません。
  - 管理者特権が必要なアプリケーションは、ローカルアプリケーションアクセスでは起動できません。
  - 単一インスタンスアプリケーション (Windows Media Playerなど) もほかのアプリケーションと同等に処理されます。
  - ローカルアプリケーションはローカルマシンのWindowsテーマで表示されます。
  - 全画面アプリケーションはサポートされません。これには、PowerPointのスライドショーやデスクトップ全体で表示されるフォトビューアーなど、全画面で開くアプリケーションが含まれます。
  - ローカルアプリケーションアクセスでは、ローカルアプリケーションのプロパティ (デスクトップや [スタート] メニューへのショートカットの配置など) が複製されます。ただし、ショートカットキーや読み取り専用属性などのほかのプロパティは複製されません。
  - 一部のアプリケーションで、各ウィンドウが正しい重なり順で表示されない場合があります。これにより、一部のウィンドウが非表示になることがあります。
  - マイコンピューター、ごみ箱、コントロールパネル、ネットワークドライブ、フォルダーなどのショートカットはサポートされません。
  - カスタムのファイルタイプ、関連付けられたプログラムのないファイル、ZIPファイル、および隠しファイルはサポートされません。
  - ビット数の異なるローカルアプリケーションとVDAアプリケーションのタスクバーでのグループ化はサポートされません。たとえば、32ビットのローカルアプリケーションと64ビットのVDAアプリケーションは、タスクバーでグループ化

されません。

- アプリケーションはCOMを使って起動できません。たとえば、Officeアプリケーション内に埋め込まれているOfficeドキュメントをクリックしても、プロセス起動が検出されないため、ローカルアプリケーション統合に失敗します。
- 明示的なURLリダイレクトのみがサポートされます。つまり、Webブラウザのアドレスバーに表示されるURLや、ブラウザ内ナビゲーションによるURLだけが正しくリダイレクトされます。
- URLリダイレクトはデスクトップセッションでのみ機能し、アプリケーションセッションでは機能しません。
- VDAセッションのローカルデスクトップフォルダーにユーザーが新しいファイルを作成することはできません。
- ローカルアプリケーションの複数のインスタンスのタスクバーアイコンは、仮想デスクトップのタスクバー設定に基づいて表示されます。ただし、ローカルで実行されているアプリケーションのショートカットは、このアプリケーションの実行インスタンスのアイコンとはグループ化されません。また、ホストされているアプリケーションの実行インスタンスやそのアプリケーションのピン留めアイコンともグループ化されません。タスクバー上のアイコンでは、ローカルで実行されているアプリケーションのウィンドウのみを閉じることができます。ローカルアプリケーションのショートカットをデスクトップタスクバーや [スタート] メニューに固定することもできますが、そのショートカットからアプリケーションを起動できなくなる場合があります。

ローカルアプリケーションアクセスは、Windows上で次のように動作します。

- Windows 8およびWindows Server 2012のショートカットの動作
  - クライアント上にインストールされたWindowsストアアプリケーションは、ローカルアプリケーションアクセスのショートカットとして列挙されません。
  - 通常、イメージファイルとビデオファイルは、デフォルトでWindowsストアアプリケーションで開かれます。ただし、ローカルアプリケーションアクセスでは、Windowsストアアプリケーションが列挙され、ショートカットがデスクトップアプリケーションで開かれます。
- Local Programsフォルダー
  - Windows 7の場合、 [スタート] メニューにLocal Programsフォルダーが表示されます。
  - Windows 8の場合、ユーザーがスタート画面のカテゴリとして [すべてのアプリ] を選択した場合のみ、Local Programsフォルダーが表示されます。Local Programsフォルダーにすべてのサブフォルダーが表示されるわけではありません。
- アプリケーション用のWindows 8グラフィック機能
  - デスクトップアプリケーションはデスクトップ領域に制限され、スタート画面およびWindows 8スタイルアプリケーションの背面に表示されます。
  - ローカルアプリケーションアクセスは、マルチモニターモードでデスクトップアプリケーションのように動作しません。マルチモニターモードでは、スタート画面とデスクトップは別のモニター上で表示されます。
- Windows 8およびローカルアプリケーションアクセスのURLリダイレクト
  - Windows 8上のInternet Explorerではアドオンを使用できないため、URLリダイレクトを有効にする場合はデスクトップ版のInternet Explorerを使用する必要があります。
  - Windows Server 2012上のInternet Explorerでは、デフォルトでアドオンが無効になっています。URLリダイレクトを実装するには、Internet Explorerの拡張構成を無効にしてください。標準ユーザーに対してアドオンが有効になるように、Internet Explorerのオプションを再設定して再起動します。

# ローカルアプリケーションアクセスとURLリダイレクトの構成

Jan 04, 2017

Citrix ReceiverでローカルアプリケーションアクセスとURLリダイレクトを使用するには：

- ローカルクライアントマシンにReceiverをインストールします。Receiverのインストール時に両方の機能を有効にしたり、グループポリシーエディターを使ってローカルアプリケーションアクセステンプレートを有効にしたりできます。
- ポリシーの [ローカルアプリケーションのアクセスを許可する] 設定で [許可] を選択します。またURLリダイレクトのため、URLホワイトリストとブラックリストポリシー設定を構成できます。詳しくは、「[ローカルアプリケーションアクセスのポリシー設定](#)」を参照してください。

すべてのローカルアプリケーションに対してローカルアプリケーションアクセスとURLリダイレクトを有効にするには

1. ポリシーの [ローカルアプリケーションのアクセスを許可する] 設定で [許可] を選択します。これにより、管理者が公開したアプリケーションおよびローカルアプリケーションアクセスのショートカットが有効かどうかクライアントにより検出されます。 [禁止] を選択すると、管理者が公開したアプリケーションとローカルアプリケーションアクセスのショートカットが機能しなくなります。このポリシー設定は、URLリダイレクトのポリシー設定と同様にマシン全体に適用されます。
2. Citrix Receiverのインストール時に、マシンのすべてのユーザーに対してローカルアプリケーションアクセスとURLリダイレクトを有効にします。この操作により、URLリダイレクト機能で使用されるWebブラウザアドオンも登録されます。コマンドプロンプトで適切なコマンドを実行して、次のオプションと共にReceiverをインストールします。

```
CitrixReceiver.exe /ALLOW_CLIENTHOSTEDAPPSURL=1
```

```
CitrixReceiverWeb.exe /ALLOW_CLIENTHOSTEDAPPSURL=1
```

1. gpedit.mscを実行します。
2. [コンピューターの構成] を選択します。 [管理用テンプレート] を右クリックし、 [テンプレートの追加と削除] > [追加] の順に選択します。
3. ReceiverのConfigurationフォルダー（通常はc:\Program Files (x86)\Citrix\Online Plugin\Configuration）にあるicaclient.admテンプレートを追加します（icaclient.admテンプレートをコンピューターのConfigurationに追加した後、 [ユーザーの構成] でそれを構成できます）。
4. [管理用テンプレート]、 [従来の管理用テンプレート（ADM）]、 [Citrix Components]、 [Citrix Receiver]、 [User authentication] の順に開きます。
5. [Local App Access settings] を選択します。
6. [有効] を選択し、 [Allow URL Redirection] チェックボックスをオンにします。URLリダイレクト機能を使用するには、以下を参照にしながらコマンドラインを使用してWebブラウザにアドオンを登録する必要もあります。

公開アプリケーションへのアクセスのみを提供するには

1. Delivery Controllerがインストールされたサーバー上で、regedit.exeを実行します。
  1. HKEY\_LOCAL\_MACHINE\Software\Wow6432Node\Citrix\DesktopStudioを開きます。
  2. REG\_DWORD値ClientHostedAppsEnabledを追加して、1を設定します（0を設定するとローカルアプリケーションアクセスが無効になります）。
2. Delivery Controllerサーバーを再起動して、Studioを起動します。

3. ローカルアプリケーションアクセスのアプリケーションを公開します。
  1. Studioのナビゲーションペインで [デリバリーグループ] を選択し、 [アプリケーション] タブを選択します。
  2. [操作] ペインで、 [ローカルアクセスアプリケーションの作成] を選択します。
  3. デスクトップのデリバリーグループを選択します。
  4. ユーザーのローカルマシン上にあるアプリケーションの実行可能ファイルのフルパスを入力します。
  5. このローカルアプリケーションのショートカットを仮想デスクトップの [スタート] メニューやデスクトップに追加するかどうかを指定します。
  6. [名前] ページでデフォルトの値をそのまま使用して、設定内容を確認します。
4. Citrix Receiverのインストール時に、マシンのすべてのユーザーに対してローカルアプリケーションアクセスとURLリダイレクトを有効にします。この操作により、URLリダイレクト機能で使用されるWebブラウザアドオンも登録されます。コマンドプロンプトで以下のコマンドを実行してReceiverをインストールします。  
 CitrixReceiver.exe /ALLOW\_CLIENTHOSTEDAPPSURL=1  
 CitrixReceiverWeb.exe /ALLOW\_CLIENTHOSTEDAPPSURL=1
5. ポリシーの [ローカルアプリケーションのアクセスを許可する] 設定で [許可] を選択します。これにより、管理者が公開したアプリケーションおよびローカルアプリケーションアクセスのショートカットが有効かどうかクライアントにより検出されます。 [禁止] を選択すると、管理者が公開したアプリケーションとローカルアプリケーションアクセスのショートカットが機能しなくなります。

注：URLリダイレクト機能で使用されるWebブラウザアドオンは、コマンドラインでReceiverをインストールするときに/ALLOW\_CLIENTHOSTEDAPPSURL=1オプションを指定すると自動的に登録されます。

以下のコマンドを実行して、適切なWebブラウザにアドオンを登録したり登録解除したりできます。

- クライアントデバイス上でアドオンを登録する場合：<client-installation-folder>\redirector.exe /reg<browser>
- クライアントデバイス上でアドオンを登録解除する場合：<client-installation-folder>\redirector.exe /unreg<browser>
- VDA上でアドオンを登録する場合：<VDAinstallation-folder>\VDARedirector.exe /reg<browser>
- VDA上でアドオンを登録解除する場合：<VDAinstallation-folder>\VDARedirector.exe /unreg<browser>

ここで、<browser>にはIE、FF、Chrome、またはAllを指定します。

たとえば、Receiverを実行するデバイス上でInternet Explorer用のアドオンを登録するには、次のコマンドを実行します。

C:\Program Files\Citrix\ICA Client\redirector.exe/regIE

また、VDA for Windows Server OSが動作するサーバー上ですべてのアドオンを登録するには、次のコマンドを実行します。

C:\Program Files (x86)\Citrix\System32\VDARedirector.exe /regAll

| 説明                                                                                                                                         | 構成                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| Internet Explorerでは、入力されたURLがデフォルトでリダイレクトされます。ブラックリストに追加されていないURLがWebブラウザやWebサイトによりほかのURLにリダイレクトされた場合、最終的なURLはブラックリストに追加されていてもリダイレクトされません。 | URLリダイレクトが正しく機能するためには、Webブラウザに表示されるメッセージに従ってアドオンを有効にする必要があります。インターネットオプションを使用するアドオンやメッセージで示されたアドオンが無効の場合、URLリダイレクトは正しく機能しません。 |
| Firefoxアドオンでは、URLが常にリダイレクトされます。                                                                                                            | Firefoxでは、アドオンのインストールを許可するかどうかを確認するメッセージが新しいタブに表示されます。URLリダイレクトが正しく機能するためには、アドオンのインストールを許可する必要があります。                          |
| Chromeのアドオンでは、ユーザーがナビゲーションによ                                                                                                               | 拡張機能が外部的にインストールされます。この拡張機能                                                                                                    |

|                                                                |                                                                                                                           |
|----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <p>説明</p> <p>開いた最終的なURL (ユーザーが入力したものでないURL) は常にリダイレクトされます。</p> | <p>構成</p> <p>を無効にすると、ChromeでURLリダイレクトが動作しなくなります。シークレットモードでURLリダイレクトを使用するには、Webブラウザの設定でシークレットモードでの拡張機能の実行を許可する必要があります。</p> |
|----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|

1. ホストされたデスクトップ上で、gpedit.mscを実行します。
  1. HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\Client Hosted Apps\Policies\Session Stateに移動します。  
64ビットシステムの場合、HKEY\_LOCAL\_MACHINE\SOFTWARE\wow6432node\Citrix\Client Hosted Apps\Policies\Session Stateに移動します。
  2. REG\_DWORD値Terminateを追加して、以下のいずれかを値のデータとして設定します。
    - 1— ユーザーが仮想デスクトップからログオフまたは切断しても、ローカルアプリケーションは実行したまま保持されます。仮想デスクトップに再接続すると、そのローカルアプリケーションが再統合されます (ローカル環境で使用可能な場合)。
    - 3— ユーザーが仮想デスクトップからログオフまたは切断した場合、ローカルアプリケーションが終了します。

# サーバーVDI

Jan 04, 2017

サーバーVDI (Virtual Desktop Infrastructure) 機能を使用すると、サーバーオペレーティングシステムからユーザーにデスクトップを配信できます。

- エンタープライズ管理者は、エンジニアやデザイナーなどのユーザーにサーバーオペレーティングシステムをVDIデスクトップとして配信できます。
- サービスプロバイダーは、クラウドからデスクトップを提供できます。これらのデスクトップは、Microsoft Services Provider License Agreement (SPLA) に準拠します。

デスクトップエクスペリエンス拡張のCitrixポリシー設定を使用すると、サーバーオペレーティングシステムでWindows 7の外観を持つデスクトップを提供できます。

サーバーVDIでは、次の機能を使用できません。

- Personal vDisk
- HDX 3D Pro
- ホストされるアプリケーション
- ローカルアプリケーションアクセス
- 直接 (非仲介) デスクトップ接続
- リモートPCアクセス

スキャナのようなTWAINデバイスと連携するサーバーVDIには、Windows Serverのデスクトップエクスペリエンス機能をインストールする必要があります。これは、Windows Server 2012のオプション機能であり、[管理ツール] > [サーバー マネージャ] > [機能] > [機能の追加] > [デスクトップ エクスペリエンス] でインストールします。

サーバーVDIは、VDA for Windows Server OSと同じサーバーオペレーティングシステムでサポートされます。

1. Windowsサーバーでインストールの準備を行います。リモートデスクトップサービスの役割サービスがインストールされていないこと、およびユーザーのセッションが1つに制限されていることを確認します。
  - Windowsサーバーマネージャーを使って、リモートデスクトップサービスの役割サービスがインストールされていないことを確認します。既にインストールされている場合は、それを削除します。
  - [1ユーザーにつき1セッションに制限する] プロパティが有効であることを確認します。
    - Windows Server 2008 R2では、[管理ツール] > [リモートデスクトップサービス] > [リモートデスクトップセッションホストの構成] の順に選択します。[設定の編集] > [全般] で、[1ユーザーにつき1セッションに制限する] 設定が[はい] であることを確認します。
    - Windows Server 2012で、レジストリを編集してターミナルサーバー設定を設定します。レジストリキー、`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TerminalServer\DWORD fSingleSessionPerUser`を1に設定します。
2. Windows Server 2008 R2では、VDAをインストールする前に、サーバー上にMicrosoft .NET Framework 3.5 SP1をインストールします。
3. コマンドラインインターフェイスで`quiet`および`/servervdi`オプションを指定して、VDAをサーバーまたはサーバーマスターイメージ上にインストールします。デフォルトでは、サーバーオペレーティングシステム上へのVDA for Windows Desktop OSのインストールはブロックされますが、コマンドラインを使うことでこれを回避できます。

```
XenDesktopVdaSetup.exe /quiet /servervdi
```

`/controllers` オプションを使用すると、コマンドラインを使ってVDAをインストール中にDelivery ControllerまたはControllerを指定できます。

ファイアウォールが手動で構成されていない限りは、`/enable_hdx_ports`オプションを使ってファイアウォールでポートを開きます。

イメージにVDAをインストールしており、MCSを使用してそのイメージからサーバー仮想マシンを作成する場合は、`/masterimage`オプションを追加します。

`/baseimage`、`/enable_hdx_3d_pro`、または`/xa_server_location`など、サーバーVDIでサポートされていない機能のオプションを指定しないでください。
4. サーバーVDIのマシンカタログを作成します。
  1. [オペレーティングシステム] ページで、[WindowsデスクトップOS] を選択します。
  2. [概要] ページで、管理者がサーバーVDI用のマシンカタログを識別できるようにマシンカタログ名と説明を指定します。これは、StudioにおいてそのマシンカタログがサーバーVDI用であることを示す唯一のインジケーターになります。VDAはサーバー上にインストールされていますが、Studioで検索すると、このサーバーVDIカタログは[デスクトップOSマシン] タブに表示されます。

5. デリバリーグループを作成し、前の手順で作成したサーバーVDIカタログを割り当てます。

VDAのインストール中にDelivery Controllerを指定しなかった場合、Citrixポリシー設定やActive Directoryを使って、あるいはVDAマシンのレジストリ値を編集して、あとから指定します。

# コンポーネントの削除

Jan 04, 2017

製品のコンポーネントを削除するには、プログラムの削除（アンインストール）や変更を行うWindowsの機能を使用することをお勧めします。または、コマンドラインや、インストールメディアに収録されているスクリプトを使用してコンポーネントをアンインストールすることもできます。

コンポーネントをアンインストールしても、そのコンポーネントと一緒にインストールされたサードパーティ製ソフトウェアはアンインストールされず、ファイアウォール設定も変更されません。Controllerをアンインストールしても、SQL Serverソフトウェアおよびデータベースは削除されません。

Controllerをアンインストールする前に、サイトからそのControllerを削除してください。StudioやDirectorをアンインストールする場合は、それらを閉じておくことをお勧めします。

Controllerをアップグレードする前の環境でWeb Interfaceを使用していた場合は、Web Interfaceコンポーネントを別途アンインストールする必要があります。この製品のインストーラーを使ってWeb Interfaceをアンインストールすることはできません。

プログラムの削除や変更を行うWindowsの機能を使用してコンポーネントをアンインストールするには

プログラムの削除や変更を行うためのWindows機能（コントロールパネルの「プログラムと機能」など）を開き、以下の操作を行います。

- Controller、Studio、Director、ライセンスサーバー、またはStoreFrontをアンインストールするには、[Citrix XenApp <version>] または [Citrix XenDesktop <version>] を選択してから右クリックし、[アンインストール] を選択します。インストーラーが起動したら、アンインストールするコンポーネントを選択します。  
StoreFrontは、[Citrix StoreFront] を右クリックしてから [アンインストール] を選択して削除することもできます。
- VDAをアンインストールするには、[Citrix Virtual Delivery Agent <version>] を選択し、右クリックしてから [アンインストール] を選択します。インストーラーが起動したら、アンインストールするコンポーネントを選択します。
- Universal Print Serverをアンインストールするには、[Citrix Universal Print Server] を選択してから右クリックし、[アンインストール] を選択します。

コマンドラインを使ってコアコンポーネントをアンインストールするには

インストールメディアの\x64\XenDesktop Setupディレクトリから、XenDesktopServerSetup.exeコマンドを実行します。

- 特定のコンポーネントのみをアンインストールするには、/removeおよび/componentsオプションを使用します。
- すべてのコンポーネントをアンインストールするには、/removeallオプションを使用します。

コマンドおよびパラメーターについては、「[コマンドラインを使ったインストール](#)」を参照してください。

たとえば、Studioをアンインストールするには次のコマンドを実行します。

```
\x64\XenDesktop Setup\XenDesktopServerSetup.exe /remove /components studio
```

コマンドラインを使ってVDAをアンインストールするには

インストールメディアの\x64\XenDesktop Setupディレクトリから、XenDesktopVdaSetup.exeコマンドを実行します。

- 特定のコンポーネントのみをアンインストールするには、/removeおよび/componentsオプションを使用します。
- すべてのコンポーネントをアンインストールするには、/removeallオプションを使用します。

コマンドおよびパラメーターについては、「[コマンドラインを使ったインストール](#)」を参照してください。

たとえば、VDAおよびReceiverをアンインストールするには次のコマンドを実行します。



```
\x64\XenDesktop Setup\XenDesktopVdaSetup.exe /removeall
```

Active Directoryのスクリプトを使ってVDAをアンインストールする方法については、[スクリプトを使用したVirtual Delivery Agentのインストールと削除](#)を参照してください。

# アップグレードと移行

Jan 04, 2017

## アップグレード

新しいバージョンのマシンやサイトをセットアップしなくても、既存の環境をアップグレードすることで最新バージョンのコンポーネントを使用できます。これをインプレースアップグレードと呼びます。次のアップグレードが可能です。

- XenDesktop 5 (以降) からXenDesktop 7.6へのアップグレード
- XenApp 7.5からXenApp 7.6へのアップグレード

XenApp 6.5ワーカーのサーバーをXenApp 7.6のVDA for Windows Server OSにアップグレードすることもできます。これは、XenApp 6.5からの移行時に補足的に行います。

XenDesktop 5 (以降) ファームまたはXenApp 7.5サイトをアップグレードするには、以下の操作を行います。

1. コアコンポーネントおよびVDAのインストール先のマシン上でインストーラーを実行します。アップグレードが可能かどうかを検出され、新しいバージョンのソフトウェアがインストールされます。
2. アップグレードしたStudioを使用して、データベースとサイトをアップグレードします。

詳しくは、「[環境のアップグレード](#)」を参照してください。

Controller Hotfixのインストールについて詳しくは、Knowledge Centerの[CTX201988](#)を参照してください。

XenApp 6.5ワーカーをXenApp 7.6 VDAにアップグレードするには、以下の操作を行います。

1. XenApp 6.5ワーカー上でXenApp 7.6インストーラーを実行します。XenApp 6.5ファームからそのワーカーが削除され、XenApp 6.5ソフトウェアがアンインストールされた後、XenApp 7.6 VDA for Windows Server OSがインストールされます。
2. アップグレードが完了したら、そのサーバーをXenApp 7.6サイトのマシンカタログとデリバリーグループに追加します。

詳しくは、「[XenApp 6.5ワーカーのVDA for Windows Server OSへのアップグレード](#)」を参照してください。

## 移行

以前のバージョンの環境から、最新バージョンの環境にデータを移行できます。XenApp 6.5またはXenDesk 4の環境を移行できます。移行処理により、XenApp/XenDesktop 7.6コンポーネントのインストール、新しいサイトの作成、既存のファームからのデータのエクスポート、および新しいサイトへのデータのインポートが行われます。

XenApp 6.5からの移行では、以下の操作を行います。

1. コアコンポーネントをインストールして、新しいXenAppサイトを作成します。
2. XenApp 6.5コントローラーで、PowerShellコマンドレットを実行してポリシーやファームのデータをXMLファイルとしてエクスポートします。エクスポートしたXMLファイルは、必要に応じてインポート用に編集できます。
3. 新しいXenApp 7.6のサイトで、PowerShellコマンドレットを実行してXMLファイルからポリシーやアプリケーションのデータをサイトにインポートします。
4. 新しいサイトで移行後のタスクを完了します。

詳しくは、「[XenApp 6.xからの移行](#)」を参照してください。

XenDesktop 4からの移行では、以下の操作を行います。

1. コアコンポーネントをインストールして、新しいXenDesktopサイトを作成します。
2. XenDesktop 4ファームで、エクスポートコマンドツールを使用してファームのデータをXMLファイルとしてエクスポートします。エクスポートしたXMLファイルは、必要に応じてインポート用に編集できます。

3. 新しいXenDesktop 7.6のサイトで、インポートコマンドツールを使用してXMLファイルからファームのデータをサイトにインポートします。
4. 新しいサイトで移行後のタスクを完了します。

詳しくは、「[XenDesktop 4からの移行](#)」のドキュメントを参照してください。

# 環境のアップグレード

Jan 04, 2017

新しいバージョンのマシンやサイトをセットアップする前に、一部の環境をアップグレードすることができます。これをインプレースアップグレードと呼びます。次のアップグレードが可能です。

- XenDesktop Version 5 (以降) から最新リリース (現在) のXenDesktopへのアップグレード
- XenApp Version 7.5 (以降) から最新リリース (現在) のXenAppへのアップグレード

XenApp 7.6のインストーラーでは、XenApp 6.5ワーカーのサーバーをXenApp 7.6のVDA for Windows Server OSにアップグレードすることもできます。XenApp 6.5の移行について詳しくは、「[XenApp 6.5ワーカーのVDA for Windows Server OSへのアップグレード](#)」を参照してください。

アップグレードを開始するには、新しいバージョンのインストーラーを実行して、既存のコアコンポーネント (Delivery Controller、Citrix Studio、Citrix Director、Citrixライセンスサーバー) とVDAをアップグレードします。アップグレードが必要なコンポーネントが自動的に特定され、それらがアップグレードされます。各コンポーネントをアップグレードしたら、アップグレードしたStudioを使ってサイトデータベースとサイトをアップグレードします。

ここで製品という語は、特に明記されていない限り、XenApp 7.xまたはXenDesktop 7.xを指します。

以下のアップグレードはサポートされていません。

- Early Release版またはTechnology Preview版からのアップグレード
- バージョン7.5よりも前のバージョンのXenAppからのアップグレード (サーバー上のXenApp 6.5ソフトウェアを新しいVDA for Windows Server OSに置き換える場合を除く。「[XenApp 6.xからの移行](#)」参照)
- バージョン5.6よりも前のバージョンのXenDesktopからのアップグレード (「[XenDesktop 4からの移行](#)」参照)
- XenAppからXenDesktopへ、またはXenDesktopからXenAppへのアップグレード

## アップグレードできる製品コンポーネントのバージョン

製品のインストーラーとStudioを使って、以下のコンポーネントをアップグレードできます。

- Delivery Controller 5以降
- VDA 5.0 SP1以降
  - 以前のVDAリリースとは異なり、VDAをアップグレードするには製品のインストーラーを使用する必要があります。MSIは使用できません。
  - マシンでReceiver for Windows (Receiver.exe) が検出された場合は、製品のインストールメディアに含まれているバージョンにアップグレードされます。
  - マシンでReceiver for Windows Enterprise (CitrixReceiverEnterprise.exe) が検出された場合は、Receiver for Windows Enterprise 3.4にアップグレードされます。
- Director 1以降
- データベース - スキーマがアップグレードされ、サイトデータベースのデータが移行されます (Version 7.xからのアップグレードでは、構成ログデータベースと監視データベースのデータも移行されます)
- Personal vDisk

適切なドキュメントの指示に従って、次の機能または製品を必要に応じてアップグレードします。

- Provisioning Services (XenApp 7.xおよびXenDesktop 7.xでは最新リリースのバージョンを使用することをお勧めします。Provisioning Services 7.0以降がサポートされます)。
  - ローリングアップグレードを使用してProvisioning Servicesサーバーをアップグレードし、vDiskのバージョン管理機能を使用してクライアントをアップグレードします。
  - Provisioning Services 7.xはXenDesktop 5でのデスクトップの作成をサポートしていません。既存のデスクトップは引き続き使用できますが、XenDesktopをアップグレードするまではProvisioning Services 7.xで新しいデスクトップを作成す

ることはできません。このため、XenDesktop 7.xとXenDesktop 5.6のサイトを同時に運用する場合は、Provisioning Services 7にアップグレードしないでください。

- Microsoft System Center Virtual Machine Manager (SCVMM)。現在の製品はSCVMM 2012およびSCVMM 2012 SP1をサポートします。XenDesktop 5.xはそれより前のバージョンをサポートします。ダウンタイムの発生を避けるには、次の順序でアップグレードします。
  1. XenDesktop 5.6 FP1より前のバージョンを実行しているControllerがある場合は、それらをXenDesktop 5.6 FP1にアップグレードします（そのバージョンのXenDesktopのドキュメントを参照してください）。
  2. SCVMMサーバーをSCVMM 2012にアップグレードします。手順については、Microsoft社のドキュメントを参照してください。
  3. XenDesktopコンポーネントを最新バージョンにアップグレードします。
  4. 必要に応じて、SCVMMサーバーをSCVMM 2012 SP1にアップグレードします。
- StoreFront。

#### 要件、制限、および準備

|   |                                                                                                                                                                                                               |
|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| • |                                                                                                                                                                                                               |
|   | 製品のインストーラーのグラフィカルインターフェイスまたはコマンドラインインターフェイスを使ってコアコンポーネントとVDAをアップグレードする必要があります。以前のバージョンからデータをインポートまたは移行することはできません。                                                                                             |
|   | 新しいバージョンのコンポーネントをインストールしたりアップグレードしたりする場合、別のマシン上のほかのアップグレードが必要なコンポーネントをアップグレードしないと、Studioにメッセージが表示されます。<br><br>たとえば、ControllerとStudioの新しいバージョンがリリースされた場合、Controllerだけをアップグレードすると古いバージョンのStudioでサイトを管理できなくなります。 |
|   | Citrixライセンスサーバーをアップグレードする前に、Subscription Advantage日が新しい製品バージョンに対して有効であることを確認してください。<br><br>製品のVersion 7.xからのアップグレードでは、この日付が2013.0522以降である必要があります。                                                             |
|   | XenDesktop Express Editionをアップグレードすることはできません。現在サポートされているエディションのライセンスを入手してインストールしてからアップグレードしてください。                                                                                                             |
|   | アップグレードを開始する前に、 <a href="#">CTX135207</a> の説明に従ってデータベースをバックアップしてください。これにより、データベースのアップグレード後に問題が発生した場合に元の状態に復元することができます。                                                                                        |
|   | 必要な場合は、テンプレートをバックアップしてハイパーバイザーをアップグレードします。                                                                                                                                                                    |
|   | 以前のバージョンのサイトを引き続き運用する必要がある場合は、 <a href="#">「混在環境に関する考慮事項」</a> を参照してください。                                                                                                                                      |
|   | VDAがWindows XPまたはWindows Vistaのマシンにインストールされている場合は、 <a href="#">「Windows XPまたはWindows Vistaを実行しているマシンのVDA」</a> を参照してください。                                                                                      |

|                                                                                                                          |
|--------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>一部のVDAを最新バージョンにアップグレードしない場合は、「<a href="#">VDAの混在環境のサポート</a>」を確認してください。</li> </ul> |
| Web Interfaceを使用していた環境では、StoreFrontを使用することをお勧めします。                                                                       |
| 製品コンポーネントをアップグレードするには、ドメインユーザーであることに加えて、そのマシンのローカル管理者である必要があります。                                                         |
| アップグレードするときには、製品 (XenAppまたはXenDesktop) は選択または指定しません (初回インストール時に設定されています)。                                                |
| サイトの停止に備えるために、後述のアップグレードの順序を確認してください。                                                                                    |

## 混在環境に関する考慮事項

環境内に異なる製品バージョンのSites/farmsがある場合 (混在環境)、StoreFrontを使って異なる製品バージョンのアプリケーションやデスクトップ (XenDesktop 7.1サイトとXenDesktop 7.5サイトがある場合など) を集約することをお勧めします。詳しくは、StoreFrontのドキュメントを参照してください。

- 通常、最新バージョンのStudioとDirectorではそのバージョンのサイトのみを管理/監視します (このバージョンのDirectorでXenDesktop 5.xのVDAを監視することは可能ですが、ログオン処理時間などのデータは使用できません)。たとえば、Version 7.6のStudioでXenDesktop 7.1のサイトを管理することはできません。同様に、Version 7.1のStudioでXenDesktop 7.6のサイトを管理することもできません。
- 以前のバージョンのControllerが動作する環境で最新バージョンのVDAを使用することは可能ですが、一部の新機能を使用できない場合があります。後述の「  
— [VDAの混在環境のサポート](#)  
」を参照してください。
- Version 5.xのControllerとVersion 7.xのVDAとの混在環境は、アップグレードに伴う一時的な構成としてのみ運用してください。可能な限り、環境内のすべてのコンポーネントのアップグレードを早急に完了してください。
- 混在環境では、異なるバージョンのStudioやDirectorを同一マシン上にインストールすることはできません。
- Provisioning Servicesを使用するXenDesktop 7.xおよびXenDesktop 5.6のサイトを同時に運用する場合は、XenDesktop 7.xサイト用に新しいProvisioning Servicesを展開するか、Provisioning Servicesを最新バージョンにアップグレードし、XenDesktop 5.6サイトでのワークロードのプロビジョニングを無効にしてください。
- スタンドアロンのStudioは、新しいバージョンを使用する準備ができるまでアップグレードしないでください。

## Windows XPまたはWindows Vista上のVDA

Windows XPまたはWindows Vistaが動作するマシンにインストールされているVDAをVersion 7.xにアップグレードすることはできません。特定のHotfixが適用されたVDA 5.6 FP1を使用する必要があります。手順については、[CTX140941](#)を参照してください。以前のバージョンのVDAをVersion 7.xのサイトで実行することは可能ですが、次のような多くの機能を使用できません。

- 新しいバージョンのVDAを必要とするStudioの機能。
- Studioを使ったApp-Vアプリケーションの構成。
- Studioを使ったReceiver StoreFrontのアドレスの構成。
- Machine Creation Servicesを使用する場合のMicrosoft Windows KMSライセンスの自動サポート。[CTX128580](#)を参照してください。
- Directorでの情報の表示：
  - [ダッシュボード]、[傾向]、および [ユーザーの詳細] ビューでログオン処理時間を算出するためのログオン時間

およびログオン終了イベント。

- HDX接続時間および認証時間のログオン処理時間ブレークダウンの詳細と、プロファイルロード、GPOロード、ログオンスクリプト、および対話型セッション確立の期間の詳細。
- マシンおよび接続の障害率のいくつかのカテゴリ。
- [ヘルプデスク] ビューおよび [ユーザーの詳細] ビューのアクティビティマネージャー。

Windows XPとWindows Vistaのマシンは、サポートされているオペレーティングシステムで再イメージ化して最新のVDAをインストールすることをお勧めします。

## Windows 8.xおよびWindows 7上のVDA

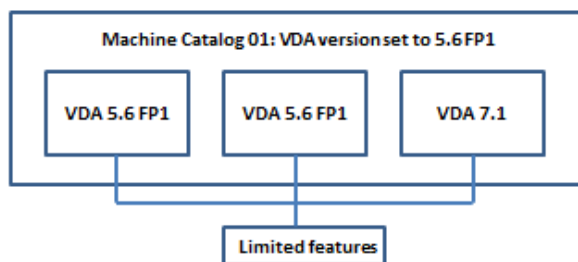
Windows 8.xまたはWindows 7を実行しているマシン上にインストールされたVDAをWindows 10にアップグレードするには、XenAppおよびXenDesktop 7.6 FP3と共に配信されたスタンドアロンVDAインストールパッケージを使って、Windows 7およびWindows 8.xマシンをWindows 10に再イメージ化して、Windows 10用にサポートされているVDAをインストールすることを推奨します。再イメージ化がオプションではない場合は、オペレーティングシステムをアップグレードする前にVDAをインストールします。これを実行しない場合、VDAは未サポート状態となります。

## VDAの混在環境のサポート

製品バージョンをアップグレードするときには、すべてのコアコンポーネントおよびVDAをアップグレードすることをお勧めします。これにより、そのエディションで追加されたり強化されたりした機能をすべて使用できるようになります。たとえば、Version 7.6の製品でサポートされるセッションの事前起動、セッション残留、および認証が不要なユーザーの機能を使用するには、Version 7.6以降のVDAがインストールされている必要があります。

環境のすべてのVDAを同時にアップグレードできない場合は、マシンカタログを作成するときに、マシンにインストールされているVDAのバージョンを指定できます。デフォルトでは、推奨される最新バージョンが指定されます。したがって、ここへ変更が必要になるのは、以前のバージョンのVDAがインストールされているマシンがマシンカタログに含まれている場合だけです。ただし、マシンカタログに複数のバージョンのVDAが混在していると、意図しない問題が生じることがあります。

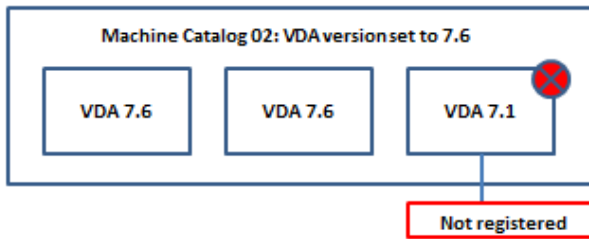
上述のとおり、環境にWindows XPおよびWindows Vistaのシステムが含まれている場合は、以前のバージョンのVDAを使用する必要があり、それらのマシンを含むマシンカタログでは、VDAのバージョンとして [5.6 FP1] を指定する必要があります。これにより、VDAがControllerに登録されますが、それらのマシンでVersion 7.xの新機能 (StoreFrontなど) を使用することはできません。これは、そのカタログにVDA 7.xがインストールされたマシンを追加した場合にも当てはまります。次の図はこのことを示しています。



上の例で、以前のバージョンのVDAを引き続き使用する必要がある場合は、それらのマシンで専用のマシンカタログを作成します。

マシンカタログを作成するときにVDAのバージョンとして推奨バージョンを指定する場合、そのカタログに以前のバージョンのVDAが含まれていると、それらのマシンはControllerに登録されず、機能しなくなります。

たとえば、VDAの最新バージョンが7.6である場合に、VDAのバージョンとしてデフォルトの [7.6 (推奨。最新機能を使用できます)] を指定してマシンカタログを作成し、VDA 7.6の2つのマシンとVDA 7.1の1つのマシンを追加したとします。

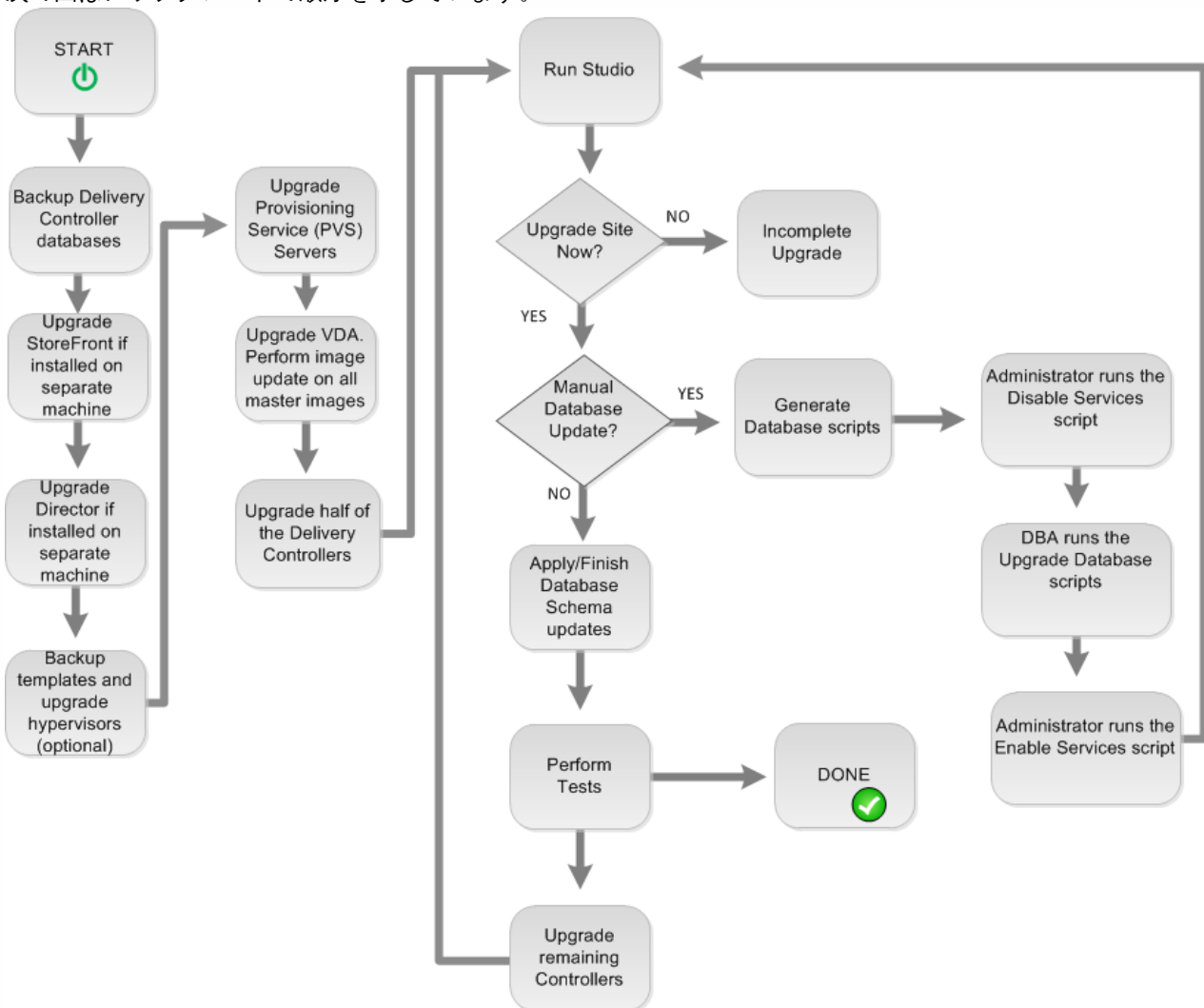


この場合、VDA 7.1のマシンはControllerに登録されません。そのVDAをアップグレードできない場合は、VDAのバージョンとして [7.0 (またはそれ以降)] を指定した別のマシンカタログを作成してそのマシンを追加します。これにより、Version 7.6の新機能は使用できませんが、そのマシンをControllerに登録できるようになります。

## アップグレードの順序

コンポーネントが複数のマシンにインストールされている場合は、各マシン上でインストーラーを実行します。

次の図はアップグレードの順序を示しています。



製品インストーラーのグラフィックインターフェイスを実行するには、マシンにログオンし、インストールメディアまたはISOファイルのAutoSelectをダブルクリックします。コマンドラインインターフェイスを使用する場合は、「  
ーコマンドラインを使ったインストール」を参照してください。

1. アップグレード可能な複数のコアコンポーネント (Controller、Studio、ライセンスサーバーなど) が同じサーバーにインストールされている場合、インストーラーによりそれらがすべてアップグレードされます。Controller以外のコアコンポー



メントがほかのマシン上にインストールされている場合は、各マシンでインストーラーを実行します。この場合、ライセンスサーバー、StoreFront、Directorの順にアップグレードすることをお勧めします。

2. Provisioning Servicesのドキュメントの指示に従って、Provisioning Servicesのサーバーとクライアントをアップグレードします。
3. 製品のインストーラーをVDAがあるマシン上で実行します。VDAのアップグレードはControllerの前でも後でも可能ですが、Controllerの前にアップグレードすることをお勧めします。これにより、アップグレード後に新しい機能をすぐに有効にできます。

物理マシン（リモートPCアクセス用のマシンも含まれます）上にインストールされているVDA 7.xからアップグレードする場合、コマンドラインインターフェイスでパラメーター「/EXCLUDE "Personal vDisk","Machine Identity Service"」を使用します。次に例を示します。

```
C:\x64\XenDesktop Setup\XenDesktopVdaSetup.exe /EXCLUDE "Personal vDisk","Machine Identity Service"
```

4. 製品のインストーラーを既存のControllerの半分で実行します（これによりこれらのサーバー上にインストールされたほかのコアコンポーネントもアップグレードされます）。たとえば、サイトにControllerが4つある場合は2つのControllerを先にアップグレードします。
  - 半分のControllerをそのまま残すことで、ユーザーはそのサイトを引き続き使用できます。VDAはこれらの残りのControllerに登録されます。動作するControllerの数が減少するため、サイトの処理能力が低下する場合があります。データベースのアップグレードの最終段階で新しいクライアント接続を確立するときに、ほんの短い間だけサイトの動作が中断されます。サイト全体がアップグレードされるまでは、アップグレード済みのControllerでは要求を処理できません。
  - サイトにControllerが1つしかない場合、アップグレード中はサイトが動作しなくなります。
5. 前の手順でアップグレードしたControllerとは別のマシンにStudioがインストールされている場合は、そのマシン上でインストーラーを実行します。
6. アップグレードしたStudioを使ってサイトデータベースをアップグレードします。詳しくは、[データベースサイトのアップグレード](#)を参照してください。
7. アップグレードしたStudioのナビゲーションペインで[Citrix Studio（サイト名）]を選択し、[よく使用するタスク]タブの[残りのDelivery Controllerのアップグレード]を選択します。
8. アップグレードが完了したら、Studioをいったん閉じてから再度開きます。
9. [よく使用するタスク]ページの[サイト構成]セクションで、[登録の実行]を選択します。Controllerに登録すると、サイトで使用できるようになります。
10. アップグレードの完了時に[完了]を選択すると、Citrixのカスタマーエクスペリエンス向上プログラム（CEIP）に登録するかどうかを選択できるページが表示されます。CEIPでは、使用環境に関する情報が匿名で収集されます。収集された情報は、弊社製品の品質、信頼性、およびパフォーマンスの向上のために使用させていただきます。
11. コンポーネント、データベース、およびサイトサイトのアップグレードが完了したら、Studioで以下のタスクを実行します。
  - アップグレードしたサイトをテストします。Studioのナビゲーションペインで[Citrix Studio（サイト名）]を選択し、[よく使用するタスク]タブの[サイトのテスト]を選択します。これらのテストはデータベースのアップグレード後に自動的に実行されますが、必要に応じて再実行できます。
  - アップグレードしたVDAを使用するマスターイメージをすべて更新します。
  - マシンカタログとデリバリーグループをアップグレードします。

## データベースとサイトのアップグレード

コアコンポーネントとVDAをアップグレードしたら、アップグレードしたStudioを使ってデータベースとサイトの自動または手動アップグレードを開始します。

- データベースの自動アップグレードでは、SQL Serverデータベーススキーマを更新できる権限（db\_securityadminまたは

db\_ownerデータベースロールなど) がStudioユーザーに必要です。

- Studioユーザーにこれらの権限がない場合は、データベースの手動アップグレードで生成されるスクリプトを使用します。生成されるスクリプトの一部をStudioユーザーがStudioから実行して、ほかスクリプトをデータベース管理者がSQL Server Management Studioなどのツールを使って実行します。SQLスクリプトを手動で実行する場合は、SQLCMDユーティリティあるいはSQLCMDモードのSQL Management Studioのいずれかを使って実行する必要があります。これ以外の場合は不正確なエラーが生じることとなります。

**重要：**データベースをアップグレードする前にバックアップすることを強くお勧めします。手順については、[CTX135207](#)を参照してください。

データベースのアップグレード中は製品サービスが無効になります。その間はControllerでサイトへの接続要求を仲介できなくなるため、慎重な計画が必要です。

データベースのアップグレードが完了し、製品サービスが有効になると、Studioにより環境と構成がテストされてHTMLレポートが生成されます。問題が見つかった場合は、データベースのバックアップを復元できます。問題を解決した後で、データベースのアップグレードを再実行します。

**データベースとサイトの自動アップグレード-** アップグレードしたStudioを起動します。[サイトの自動アップグレードを開始する]を選択して準備内容を確認すると、データベースとサイトのアップグレードが開始されます。

**データベースとサイトの手動アップグレード-** この方法ではスクリプトを生成して実行します。

1. アップグレードしたStudioを起動します。[このサイトを手動でアップグレードする]を選択すると、ウィザードによってライセンスサーバーの互換性がチェックされて確認メッセージが表示されます。データベースをバックアップしてあることを確認すると、スクリプトとアップグレード手順のチェックリストが生成されて表示されます。
2. 以下のスクリプトを順番に実行します。

| スクリプト                      | 説明                                                                                                                                    |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| DisableServices.ps1        | Controller上でStudioユーザーが実行するPowerShellスクリプト。製品サービスを無効にします。                                                                             |
| UpgradeSiteDatabase.sql    | サイトデータベースを格納しているサーバー上でデータベース管理者がSQL Server Management Studioなどのツールを使って実行するSQLスクリプト。                                                   |
| UpgradeMonitorDatabase.sql | 監視データベースを格納しているサーバー上でデータベース管理者がSQL Server Management Studioなどのツールを使って実行するSQLスクリプト。                                                    |
| UpgradeLoggingDatabase.sql | 構成ログデータベースを格納しているサーバー上でデータベース管理者がSQL Server Management Studioなどのツールを使って実行するSQLスクリプト。このスクリプトは、このデータベースが変更された場合にのみ実行します (Hotfixの適用後など)。 |
| EnableServices.ps1         | Controller上でStudioユーザーが実行するPowerShellスクリプト。製品サービスを有効にします。                                                                             |

3. 表示されたチェックリストのタスクを完了したら、[アップグレードを完了し、[よく使うタスク]に戻る]を選択します。

# XenApp 6.5ワーカーのVDA for Windows Server OSへのアップグレード

Jan 04, 2017

各XenApp 6.5ワーカー上でXenApp 7.6インストーラーを実行すると、以下の処理が行われます。

- XenApp 6.5ファームからそのワーカーが削除される（このタスクでは自動的にXenApp 6.5インストーラーのコマンドラインインターフェイスが使用されます）。
- XenApp 6.5ソフトウェアがアンインストールされる。
- 新しいVDA for Windows Server OS（XenApp 7.6またはそれ以降のバージョン）がインストールされる。

インストーラーのグラフィカルインターフェイスでインストールする場合、新しいXenAppサイトにVDA for Windows Server OSをインストールするときと同じウィザードを使用します。同様に、コマンドラインインターフェイスでのインストールでも、ほかのVDAのインストールと同じコマンドおよびパラメーターを使用します。

XenApp 7.6のコアコンポーネントおよびほかのVDAをインストールしたことがある場合は、そのときと同じ方法でインストーラーを操作します。インストールの準備について詳しくは、「[VDAのインストールの準備](#)」を参照してください。XenApp 6.5ワーカーサーバー上でインストーラーを起動（[グラフィカルインターフェイスを使ったインストール](#)）するかコマンドを実行（[コマンドラインを使ったVDAのインストール](#)）します。

ヒント：

- このアップグレードは、セッションホスト専用モード（セッションホストモードまたはワーカーとも呼ばれます）で構成されたXenApp 6.5サーバーで有効です。
- XenApp 6.5のアンインストール時に、サーバーの再起動が何回か必要です。コマンドラインでのインストールでは、/NOREBOOTオプションを指定することで自動再起動を無効にできます。ただし、アンインストール処理を完了して次のインストール処理を行うためには、サーバーを再起動する必要があります。
- XenAppのアンインストール時にエラーが発生した場合は、エラーメッセージに従ってアンインストールログを参照してください。アンインストールログは、%TEMP%\Citrix\XenDesktop Installation\XenApp 6.5 Uninstall Log Files\フォルダーに生成されます。
- XenApp 6.5ワーカーをアップグレードしたら、新しいXenAppサイトのStudioを使用してそのワーカーのマシンカタログを作成（または既存のマシンカタログを編集）します。
- XenApp 6.5のコントローラーサーバーからポリシーやアプリケーション設定を移行した場合（[XenApp 6.xからの移行](#)参照）は、移行した公開アプリケーションのデリバリーグループを、XenApp 6.5でこれらのアプリケーションをホストしていたマシンカタログに割り当てます。

## トラブルシューティング

問題：XenApp 6.5ソフトウェアのアンインストールに失敗し、アンインストールログにメッセージ「Error 25703.An error occurred while plugging XML into Internet Information Server.Setup cannot copy files to your IIS Scripts directory.Please make sure that your IIS installation is correct.」が記録される。

- 原因：この問題は、元のXenApp 6.5のインストール時にCitrix XML Service (CtxHttp.exe) とIISとのポート共有を無効にしたシステム、および.NET Framework 3.5.1がインストールされたシステムで発生します。
- 解決策：
  1. Windowsのサーバーの役割の削除ウィザードを使用して [Webサーバー (IIS) の役割] を削除します。Webサーバー (IIS) の役割は必要に応じて後で再インストールできます。
  2. サーバーを再起動します。
  3. コントロールパネルで以下のソフトウェアをアンインストールします。
    1. Citrix XenApp 6.5

2. Microsoft Visual C++ 2005 Redistributable (x64), version 8.0.56336
4. サーバーを再起動します。
5. XenApp 7.6のインストーラーを実行してVDA for Windows Server OSをインストールします。

# XenApp 6.xからの移行

Jan 04, 2017

重要：移行を開始する前に、このアークル全体をよくお読みください。

XenApp 6.xの移行ツール（以下移行ツール）では、いくつかのPowerShellスクリプトが実行されます。これらのスクリプトにより、XenApp 6.x（6.0または6.5）のポリシーおよびファームデータを移行するためのコマンドレットが実行されます。XenApp 6.xのコントローラーサーバー上でエクスポートコマンドレットを実行して、データをXMLファイルにエクスポートします。その後でXenApp 7.6 Controller上でインポートコマンドレットを実行して、エクスポートされたデータに基づいてオブジェクトを作成します。

移行ツールの概要ビデオについては[ここを参照してください](#)。

移行プロセスの主な手順は以下のとおりです。詳しい説明については後述します。

1. XenApp 6.0または6.5のコントローラー上で、以下の操作を行います。
  1. PowerShellのエクスポートモジュールをインポートします。
  2. エクスポートコマンドレットを実行して、ポリシーとファームのデータをXMLファイルにエクスポートします。
2. エクスポートしたXMLファイルおよびアイコンフォルダー（エクスポート時にアイコンをXMLファイルに埋め込まなかった場合）をXenApp 7.6のControllerにコピーします。
3. XenApp 7.6のController上で、以下の操作を行います。
  1. PowerShellのインポートモジュールをインポートします。
  2. インポートコマンドレットを実行して、ポリシーとファーム（アプリケーション）のデータをXMLファイルからインポートします。
4. 移行後のタスクを完了します。

実際に移行を実行する前に、XenApp 6.xの設定をエクスポートしてからXenApp 7.6サイトでプレビューインポートを実行できます。これにより、問題が発生する可能性がある箇所を特定して、実際にインポートを実行する前に問題を修正できます。たとえば、新しいXenApp 7.6サイトに同じ名前のアプリケーションが既に存在していることが検出される場合があります。プレビューによって生成されるログファイルを実際の移行を実行するためのガイドとして使用することができます。

特に明記されている場合を除き、6.xはXenApp 6.0または6.5を指します。

## このリリースでの新機能

2014年12月のリリース（Version 20141125）には、以下の更新が含まれています。

- XenApp 6.0ファームで移行ツールを使用するときに問題が発生する場合は、サポートフォーラム (<http://discussions.citrix.com/forum/1411-xenapp-7x/>) までご報告ください。このツールのアップデート開発にあたり、調査させていただきます。
- 新しいパッケージ - XAMigration.zipファイルには、ReadIMA.zipとImportFMA.zipという2つのパッケージが含まれています。XenApp 6.xサーバーからのエクスポートでは、ReadIMA.zipのみが必要です。XenApp 7.6サーバーへのインポートでは、ImportFMA.zipのみが必要です。
- Export-XAFarmコマンドレットにEmbedIconDataパラメーターが追加されました。これを使用すると、アイコンデータを別個のファイルとしてコピーする必要がなくなります。
- Import-XAFarmコマンドレットに以下の3つのパラメーターが追加されました。
  - MatchServer - 条件に一致する名前のサーバーからアプリケーションをインポートします。
  - NotMatchServer - 条件に一致しない名前のサーバーからアプリケーションをインポートします。
  - IncludeDisabledApps - 無効に設定されたアプリケーションをインポートします。
- 事前起動アプリケーションはインポートされません。
- Export-Policyコマンドレットは、XenDesktop 7.x上で機能します。

## 移行ツールのパッケージ

移行ツールのパッケージは、Citrix社の[ダウンロードサイト](#)から入手できます。XAMigration.zipファイルには、以下の2つのパッケージが含まれています。

- ReadIMA.zip - XenApp 6.xファームからデータをエクスポートするためのファイルと、共有モジュールが含まれています。

| モジュールまたはファイル | 説明 |
|--------------|----|
|--------------|----|

| モジュールまたはファイル      | 説明                                                      |
|-------------------|---------------------------------------------------------|
| ExportPolicy.psm1 | XenApp 6.xのポリシーをXMLファイルにエクスポートするPowerShellスクリプトモジュール。   |
| ExportXAFarm.psm1 | XenApp 6.xのファーム設定をXMLファイルにエクスポートするPowerShellスクリプトモジュール。 |
| ExportPolicy.psd1 | スクリプトモジュールExportPolicy.psm1のPowerShellマニフェストファイル。       |
| ExportXAFarm.psd1 | スクリプトモジュールExportXAFarm.psm1のPowerShellマニフェストファイル。       |
| LogUtilities.psm1 | ログ機能を含んでいる共有PowerShellスクリプトモジュール。                       |
| XmlUtilities.psd1 | スクリプトモジュールXmlUtilities.psm1のPowerShellマニフェストファイル。       |
| XmlUtilities.psm1 | XML機能を含んでいる共有PowerShellスクリプトモジュール。                      |

- ImportFMA.zip - XenApp 7.6ファームにデータをインポートするためのファイルと、共有モジュールが含まれています。

| モジュールまたはファイル      | 説明                                                  |
|-------------------|-----------------------------------------------------|
| ImportPolicy.psm1 | ポリシーをXenApp 7.6サイトにインポートするPowerShellスクリプトモジュール。     |
| ImportXAFarm.psm1 | アプリケーションをXenApp 7.6サイトにインポートするPowerShellスクリプトモジュール。 |
| ImportPolicy.psd1 | スクリプトモジュールImportPolicy.psm1のPowerShellマニフェストファイル。   |
| ImportXAFarm.psd1 | スクリプトモジュールImportXAFarm.psm1のPowerShellマニフェストファイル。   |
| PolicyData.xsd    | ポリシーデータのXMLスキーマ。                                    |
| XAFarmData.xsd    | XenAppファームデータのXMLスキーマ。                              |
| LogUtilities.psm1 | ログ機能を含んでいる共有PowerShellスクリプトモジュール。                   |
| XmlUtilities.psd1 | スクリプトモジュールXmlUtilities.psm1のPowerShellマニフェストファイル。   |
| XmlUtilities.psm1 | XML機能を含んでいる共有PowerShellスクリプトモジュール。                  |

#### 制限事項

- 一部のポリシー設定はインポートされません（「[インポートされないポリシー設定](#)」参照）。サポートされない設定は無視され、ログファイルに記録されます。
- エクスポートによりすべてのアプリケーションの詳細がXMLファイルにエクスポートされますが、サーバー上にインストールされたアプリケーションだけXenApp 7.6サイトにインポートされます。つまり、公開アプリケーション、公開コンテンツ、および多くのストリーム配信アプリケーションはインポートされません。例外については、「[手順：データのインポート](#)」のImport-XAFarmコマンドレットのパラメーターを参照してください。
- アプリケーションサーバーはインポートされません。
- XenApp 6.xのIndependent Management Architecture (IMA) とXenApp 7.6のFlexCast Management Architecture (FMA) ではアーキテクチャが異なるため、アプリケーションプロパティの多くはインポートされません（「[アプリケーションプロパティの対応](#)」参照）。
- インポートによりデリバリーグループは作成されません。インポートする項目を制限するためのパラメーターの使用については、「[高度な使用方法](#)」を参照してください。
- AppCenterで作成されたCitrixポリシーの設定だけがインポートされます。Windowsグループポリシーオブジェクト (GPO) で作成されたCitrixポリシーはインポートされません。
- 移行スクリプトは、XenApp 6.xからXenApp 7.6への移行のみを対象としています。
- Studioは複数階層で5レベルよりも深いフォルダーはサポートされず、インポートされません。アプリケーションフォルダー構造に5レベルより深いフォルダーがある場合、インポートを実行する前にフォルダーレベルの階層数を減らすことを検討してください。

## セキュリティに関する注意事項

エクスポートスクリプトによって作成されるXMLファイルには環境と組織に関する機密情報（ユーザー名、サーバー名、およびXenAppファーム、アプリケーション、ポリシーの構成データなど）が含まれることがあるため、安全な取り扱いが必要です。

ポリシーとアプリケーションをインポートする前に、XMLファイルが不正に変更されていないことを確認してください。

ポリシーの適用方法は、オブジェクトへのポリシーの割り当て（ポリシーフィルター）によって制御されます。ポリシーをインポートしたら、各ポリシーのオブジェクト割り当てをよく調べて、インポートに起因するセキュリティ上の脆弱性がないことを確認してください。インポートしたポリシーに別のユーザー、IPアドレス、またはクライアント名が割り当てられていることがあります。許可/拒否の設定の意味が変わることもあります。

## ログとエラー処理

スクリプトにより、コマンドレットのすべての実行、情報メッセージ、実行結果、警告、およびエラーについてのログが記録されます。

- ほとんどのCitrix PowerShellコマンドレットでログが生成されます。インポートスクリプトで新しいサイトオブジェクトを作成するために実行されるすべてのPowerShellコマンドレットでもログが生成されます。
- スクリプトの実行状況（処理中のオブジェクトなど）についてのログが記録されます。
- 処理フローの状態に影響する主な操作のログが記録されます。これには、コマンドラインからのフローも含まれます。
- コンソールに出力されるすべてのメッセージがログに記録されます。これには、警告やエラーも含まれます。
- ログの各行には、ミリ秒までのタイムスタンプが記録されます。

エクスポートおよびインポートのコマンドレットを実行するときは、常に個別のログファイルを指定することをお勧めします。

ログファイルの名前を指定しない場合、実行ユーザーの既存のホームフォルダー（PowerShell \$HOME変数で指定される）にログファイルが格納されます。ホームフォルダーが存在しない場合は、スクリプトの現在の実行フォルダーに格納されます。デフォルトでは、「XFarm<YYYYMMDDHHmmSS-xxxxxx>」というログファイルが生成されます。ここで、「xxxxxx」はランダムな数字で構成されます。

デフォルトでは、すべての進捗情報が表示されます。表示されないようにするには、エクスポートコマンドレットとインポートコマンドレットでNoDetailsパラメーターを指定します。

通常、スクリプトでエラーが発生すると処理がそこで停止します。エラー状態を解決した後で、コマンドレットを再実行できます。

エラーとしてみなされない状態も多くは警告としてログに記録され、スクリプトの実行は続行されます。たとえば、アプリケーションの種類がサポートされない場合は警告として記録され、インポートから除外されます。XenApp 7.6サイトに既に存在するアプリケーションはインポートされません。XenApp 7.6でサポートされなくなったポリシー設定はインポートされません。

移行スクリプトでは多くのPowerShellコマンドレットが使用され、一部のエラーがログに記録されない場合があります。ログに記録される情報を追加するには、PowerShellのログ機能を使用してください。たとえば、PowerShellで画面上に出力されるすべての情報をログに記録できます。詳しくは、Start-TranscriptおよびStop-Transcriptコマンドレットのヘルプを参照してください。

## 要件、準備、およびベストプラクティス

**重要：**移行を開始する前に、このトピック全体をよくお読みください。

PowerShellの実行ポリシー、モジュール、スクリプトなどについての基本的な理解が必要です。スクリプトを作成するための専門知識は必要ありませんが、実行するコマンドレットについて理解しておく必要があります。事前にGet-Helpコマンドレットを使用して、各移行コマンドレットのヘルプを確認しておいてください。次に例を示します。

Get-Help -full Import-XAFarm

コマンドラインでログファイルを指定して、コマンドレットの実行後にそのログファイルを確認してください。スクリプトの処理に失敗した場合は、ログファイルを参照してエラーを解決し、コマンドレットを再実行してください。

ヒント：

- 2つの環境（XenApp 6.xファームと新しいXenApp 7.6サイト）を実行している間のアプリケーション配信を容易にするには、StoreFrontまたはWeb Interfaceで両方の環境を集約します。eDocsで、使用するStoreFrontまたはWeb Interfaceのドキュメント（「管理」の「ストアの作成」）を参照してください。
- アプリケーションのアイコンデータは、以下のいずれかの方法で処理されます。
  - EmbedIconDataパラメーターを指定してExport-XAFarmコマンドレットを実行すると、アプリケーションのアイコンデータが埋め込まれたXMLファイルがエクスポートされます。
  - EmbedIconDataパラメーターを指定せずにExport-XAFarmコマンドレットを実行すると、エクスポートされたXMLファイルのベース名に「-icons」を追加した名前のフォルダーにアプリケーションのアイコンデータが格納されます。たとえば、XmlOutputFileパラメーターで「FarmData.xml」を指定した場合は、「FarmData-icons」というフォルダーが作成され、そこにアプリケーションアイコンが格納されます。このフォルダーに格納されるアイコンデータファイルは、その公開アプリケーションの表示名に基づく名前のTXTファイルです。これらのファイルにはエンコードされたバイナリアイコンデータが記述されており、インポートスクリプトがこれを読み込んでアプリケーションアイコンを再作成します。インポート時にXMLファイルと同じフォルダー内にアイコンフォルダーが見つからない場合、インポートした各アプリケーションで標準的なアイコンが使用されます。
- スクリプトモジュール、マニフェストファイル、共有モジュール、およびコマンドレットには類似した名前が付けられています。Tabキーによる補完機能

使用して、入力エラーを避けることができます。たとえば、Export-XAFarmはコマンドレット名です。ExportXAFarm.psd1およびExportXAFarm.psm1には似たファイル名が付いていますが、コマンドレットとは異なり実行不能です。

- 後述の手順説明で、<string>パラメーターの値は二重引用符で囲んであります。値にスペースが含まれない場合、二重引用符で囲まなくても正しく処理されます。

#### XenApp 6.xサーバーからのエクスポート：

- エクスポートは、コントローラーモードとセッションホストモード（通常コントローラーと呼ばれます）で構成されたXenApp 6.xサーバー上で実行する必要があります。
- エクスポートコマンドレットを実行するには、オブジェクトの読み取り権限を持つXenApp管理者である必要があります。また、PowerShellスクリプトを実行するためのWindows権限も必要です。設定方法は、後述の手順説明に記載されています。
- エクスポートを開始する前に、XenApp 6.xファームのヘルス状態が良好であることを確認します。ファームデータベースをバックアップします。Citrix IMA Helperツール（CTX137461）を使用してファームの整合性を確認します。[IMA Datastore] タブでMaster Checkを実行します（実行後、DSCheckオプションを使用して無効なエントリを解決します）。移行前に問題を解決しておく、エクスポートエラーを避けることができます。たとえば、ファームから不適切に削除されたサーバーがあると、データベースにそのサーバーのデータが残ってしまい、エクスポートスクリプトのコマンドレット（Get-XAServerZoneNameなど）の処理が失敗することがあります。コマンドレットの処理が失敗すると、スクリプトの処理も失敗します。
- ファームにユーザーが接続している状態でもエクスポートコマンドレットを実行できます。エクスポートスクリプトでは、静的なファーム構成とポリシーデータのみが読み取られます。

#### XenApp 7.6サーバーへのインポート：

- XenApp 7.6環境（およびそれ以降の移行可能なバージョン）にデータをインポートできます。XenApp 6.xファームからエクスポートしたデータをインポートする前に、XenApp 7.6のControllerとStudioをインストールしてサイトを作成しておく必要があります。設定をインポートするためにVDAをインストールしておく必要はありませんが、インストールしておくアプリケーションのファイルタイプが使用可能になります。
- インポートコマンドレットを実行するには、オブジェクトの読み取り権限および作成権限を持つXenApp管理者である必要があります。すべての管理権限を実行できる管理者には必要な権限が付与されています。また、PowerShellスクリプトを実行するためのWindows権限も必要です。設定方法は、後述の手順説明に記載されています。
- インポート時にほかのユーザーが接続していないことを確認してください。インポートスクリプトにより多くのオブジェクトが作成されるため、ほかのユーザーが同時に構成を変更すると環境が正しく機能しなくなることがあります。

エクスポートしたデータを事前にテストするために、インポートコマンドレットに-Previewパラメーターを指定して実行することができます。これにより、実際にインポートを行う前に、インポート内容を確認することができます。このパラメーターでは、実際のインポートにより実行される処理がログとして出力されます。ここでエラーが発生した場合は、それを解決してからインポートを行います。

#### 手順：データのエクスポート

エクスポート処理のビデオについては[ここを参照してください](#)。

XenApp 6.xのコントローラーからXMLファイルにデータをエクスポートするには、次の手順に従います。

1. Citrixのダウンロードサイトから、移行ツールのパッケージXAMigration.zipをダウンロードします。XenApp 6.xファームおよびXenApp 7.6サイトの両方からアクセスできるネットワーク上の共有フォルダーにパッケージをダウンロードすると便利です。ネットワーク上の共有フォルダーでXAMigration.zipを展開します。ReadIMAZipおよびImportFMAZipの2つのZIPファイルが展開されます。
2. ファームに対する読み取り以上の権限と、PowerShellスクリプトを実行できるWindows権限を持つXenApp管理者としてXenApp 6.xのコントローラーにログオンします。
3. 共有フォルダーのReadIMAZipをXenApp 6.xのコントローラーにコピーします。コントローラー上のReadIMAZipを展開し、新規フォルダー（C:\XAMigrationなど）にスクリプトを展開します。
4. PowerShellコンソールを開き、カレントディレクトリをスクリプトの場所に移動します。次に例を示します。  
cd C:\XAMigration
5. Get-ExecutionPolicyを実行して、スクリプトの実行ポリシーを確認します。
6. スクリプトを実行するには、実行ポリシーをRemoteSignedまたはそれ以上に設定する必要があります。次に例を示します。  
Set-ExecutionPolicy RemoteSigned
7. 以下のコマンドを実行して、モジュール定義ファイルExportPolicy.psd1およびExportXAFarm.psd1をインポートします。  
Import-Module .\ExportPolicy.psd1

```
Import-Module .\ExportXAFarm.psd1
```

#### ヒント：

- 移行するデータがポリシーデータの場合のみ、モジュール定義ファイルExportPolicy.psd1のみをインポートできます。同様に、移行するデータがファームデータの場合のみ、ExportXAFarm.psd1のみをインポートできます。
  - モジュール定義ファイルをインポートすると、必要なPowerShellスナップインも追加されます。
  - スクリプトモジュールファイル（\*.psm1）をインポートしないでください。
8. Export-Policyコマンドレットを実行して、ポリシーデータをエクスポートします。

| パラメーター | 説明 |
|--------|----|
|        |    |



|                                           |                                                                                                                                                                                                           |
|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメーター<br>XmlOutputFile<br>"<string>.xml" | 出力するXMLファイルの名前です。このファイルにエクスポートされたデータが記述されます。拡張子として「.xml」を指定する必要があります。既存のファイル名を指定しないでください。ただし、パスを指定する場合は既存のパスを指定してください。<br><br>デフォルト：なし。必須パラメーターです。                                                        |
| -LogFile "<br><string>"                   | ログファイルの名前です。必要に応じて拡張子を指定します。指定したファイルが存在しない場合は新規に作成されます。ファイルが存在し、-NoClobberパラメーターも指定した場合はエラーが発生します。-NoClobberパラメーターを指定しない場合は既存のファイルが上書きされます。<br><br>デフォルト：「 <a href="#">ログとエラー処理</a> 」を参照してください。           |
| -NoLog                                    | ログファイルを生成しません。このパラメーターを指定すると、-LogFileパラメーターが無視されます。<br><br>デフォルト：False。ログファイルが生成されます。                                                                                                                     |
| -NoClobber                                | -LogFileパラメーターで指定した既存のログファイルの上書きを禁止します。指定したログファイルが存在しない場合、このパラメーターは無視されます。<br><br>デフォルト：False。既存のファイルが上書きされます。                                                                                            |
| -NoDetails                                | スクリプト実行の詳細なレポートをコンソールに表示しません。<br><br>デフォルト：False。詳細なレポートがコンソールに表示されます。                                                                                                                                    |
| - SuppressLogo                            | メッセージ「XenApp 6.x to XenApp/XenDesktop 7.6 Migration Tool Version #<yyyyMMdd-hhmm>#」をコンソールに表示しません。トラブルシューティング時にスクリプトのバージョンを確認する場合に便利なため、このパラメーターを指定することは推奨されません。<br><br>デフォルト：False。上記のメッセージがコンソールに表示されます。 |

例：次のコマンドレットを実行すると、MyPolicies.xmlという名前のXMLファイルにポリシー情報がエクスポートされます。この処理は、MyPolicies.logという名前のログファイルに記録されます。

```
Export-Policy -XmlOutputFile ".\MyPolicies.XML" -LogFile ".\MyPolicies.Log"
```

9. Export-XAFarmコマンドレットを実行して、ファームデータをエクスポートします。

| パラメーター                           | 説明                                                                                                                                                                                              |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -XmlOutputFile<br>"<string>.xml" | 出力するXMLファイルの名前です。このファイルにエクスポートされたデータが記述されます。拡張子として「.xml」を指定する必要があります。既存のファイル名を指定しないでください。ただし、パスを指定する場合は既存のパスを指定してください。<br><br>デフォルト：なし。必須パラメーターです。                                              |
| -LogFile "<br><string>"          | ログファイルの名前です。必要に応じて拡張子を指定します。指定したファイルが存在しない場合は新規に作成されます。ファイルが存在し、-NoClobberパラメーターも指定した場合はエラーが発生します。-NoClobberパラメーターを指定しない場合は既存のファイルが上書きされます。<br><br>デフォルト：「 <a href="#">ログとエラー処理</a> 」を参照してください。 |
| -NoLog                           | ログファイルを生成しません。このパラメーターを指定すると、-LogFileパラメーターが無視されます。<br><br>デフォルト：False。ログファイルが生成されます。                                                                                                           |
| -NoClobber                       | -LogFileパラメーターで指定した既存のログファイルの上書きを禁止します。指定したログファイルが存在しない場合、このパラメーターは無視されます。<br><br>デフォルト：False。既存のファイルが上書きされます。                                                                                  |
| -NoDetails                       | スクリプト実行の詳細なレポートをコンソールに表示しません。<br><br>デフォルト：False。詳細なレポートがコンソールに表示されます。                                                                                                                          |

|                         |                                                                                                                                                                                                                                      |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメーター<br>-SuppressLogo | <p><b>説明</b><br/>メッセージ「XenApp 6.x to XenApp/XenDesktop 7.6 Migration Tool Version #&lt;yyyyMMdd-hhmm&gt;#」をコンソールに表示しません。トラブルシューティング時にスクリプトのバージョンを確認する場合に便利のため、このパラメーターを指定することは推奨されません。</p> <p>デフォルト：False。上記のメッセージがコンソールに表示されます。</p> |
| -IgnoreAdmins           | <p>管理者情報をエクスポートから除外します。使用方法については、<a href="#">高度な使用方法</a>を参照してください。</p> <p>デフォルト：False。管理者情報がエクスポートされます。</p>                                                                                                                          |
| -IgnoreApps             | <p>アプリケーション情報をエクスポートから除外します。使用方法については、<a href="#">高度な使用方法</a>を参照してください。</p> <p>デフォルト：False。アプリケーション情報がエクスポートされます。</p>                                                                                                                |
| -IgnoreServers          | <p>サーバー情報をエクスポートから除外します。</p> <p>デフォルト：False。サーバー情報がエクスポートされます。</p>                                                                                                                                                                   |
| -IgnoreZones            | <p>ゾーン情報をエクスポートから除外します。</p> <p>デフォルト：False。ゾーン情報がエクスポートされます。</p>                                                                                                                                                                     |
| -IgnoreOthers           | <p>そのほかの情報（構成ログ、負荷評価基準、負荷分散ポリシー、プリンタードライバー、およびワーカーグループ）をエクスポートから除外します。</p> <p>デフォルト：False。上記の情報がエクスポートされます。</p> <p>注：-IgnoreOtherスイッチの目的は、エクスポートまたはインポートに使用中の実データには影響がないエラーが生じた場合に、エクスポート処理を可能にすることです。</p>                           |
| -AppLimit<br><integer>  | <p>エクスポートするアプリケーションの数を指定します。使用方法については、<a href="#">高度な使用方法</a>を参照してください。</p> <p>デフォルト：すべてのアプリケーションがエクスポートされます。</p>                                                                                                                    |
| -EmbedIconData          | <p>アプリケーションのアイコンデータをXMLファイル内に埋め込みます。</p> <p>デフォルト：アイコンデータは埋め込まれず、別フォルダー内に格納されます。詳しくは、<a href="#">要件、準備、およびベストプラクティス</a>を参照してください。</p>                                                                                                |
| -SkipApps<br><integer>  | <p>エクスポートを省略するアプリケーションの数を指定します。使用方法については、<a href="#">高度な使用方法</a>を参照してください。</p> <p>デフォルト：アプリケーションは省略されません。</p>                                                                                                                        |

例：次のコマンドレットを実行すると、MyFarm.xmlという名前のXMLファイルにファーム情報がエクスポートされます。この処理は、MyFarm.logという名前のログファイルに記録されます。MyFarm.XMLと同じフォルダー内に「MyFarm-icons」という名前のフォルダーが作成され、そこにアプリケーションのアイコンデータが格納されます。

```
Export-XAFarm -XmlOutputFile ".\MyFarm.XML" -LogFile ".\MyFarm.Log"
```

エクスポートスクリプトの処理が完了すると、コマンドラインで指定したXMLファイルにポリシーとXenAppファームのデータが含まれています。アプリケーションアイコンファイルにはアイコンデータが含まれ、ログファイルにはエクスポート処理の内容が記述されます。

手順：データのインポート

インポート処理のビデオについては[ここを参照してください](#)。

エクスポートしたデータを事前にテストするために、Import-PolicyまたはImport-XAFarmコマンドレットに-Previewパラメーターを指定して実行することができます。これにより、実際にインポートを行う前に、インポート内容をログファイルで確認することができます。

エクスポートで生成されたXMLファイルの情報をXenApp 7.6サイトにインポートするには、次の手順に従います。

1. サイトに対する読み取り/書き込み権限と、PowerShellスクリプトを実行できるWindows権限を持つ管理者としてXenApp 7.6のControllerにログオンします。
2. 移行ツールのパッケージXAMigration.zipをネットワーク上の共有フォルダーに展開します。共有フォルダーのImportFMA.zipをXenApp 7.6のControllerにコ

ピーします。Controller上のImportFMA.zipを展開し、新規フォルダー（C:\XAMigrationなど）にスクリプトを展開します。

- XenApp 6.xのコントローラーでエクスポートされたXMLファイルを、XenApp 7.6 ControllerのImportFMA.zipの展開先フォルダー（つまりC:\XAMigrationなど）にコピーします。

Export-XAFarmコマンドレット実行時にアプリケーションのアイコンデータをXMLファイルに埋め込まなかった場合は、アイコンデータのフォルダーおよびファイルをXMLファイルのコピー先フォルダー（つまりC:\XAMigrationなど）にコピーします。

- PowerShellコンソールを開き、カレントディレクトリをスクリプトの場所に移動します。  
cd C:\XAMigration
- Get-ExecutionPolicyを実行して、スクリプトの実行ポリシーを確認します。
- スクリプトを実行するには、実行ポリシーをRemoteSignedまたはそれ以上に設定する必要があります。次に例を示します。  
Set-ExecutionPolicy RemoteSigned
- 以下のコマンドを実行して、PowerShellモジュール定義ファイルImportPolicy.psd1およびImportXAFarm.psd1をインポートします。  
Import-Module .\ImportPolicy.psd1  
  
Import-Module .\ImportXAFarm.psd1

ヒント：

- 移行するデータがポリシーデータのみの場合は、モジュール定義ファイルImportPolicy.psd1のみをインポートできます。同様に、移行するデータがファームデータのみの場合は、ImportXAFarm.psd1のみをインポートできます。
  - モジュール定義ファイルをインポートすると、必要なPowerShellスナップインも追加されます。
  - スクリプトモジュールファイル (\*.psm1) をインポートしないでください。
- Import-Policyコマンドレットを実行して、ポリシーデータをインポートします。

| パラメーター                          | 説明                                                                                                                                                                                                        |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -XmlInputFile<br>"<string>.xml" | 入力するXMLファイルの名前です。Export-Policyコマンドレットで収集されたデータが記述されています。拡張子として「.xml」を指定する必要があります。<br><br>デフォルト：なし。必須パラメーターです。                                                                                             |
| -XsdFile "<br><string>"         | XSDファイルの名前です。インポートスクリプトでは、このファイルにより入力XMLファイルの構文が検証されます。使用方法については、「 <a href="#">高度な使用方法</a> 」を参照してください。<br><br>デフォルト：PolicyData.XSD。                                                                       |
| -LogFile "<br><string>"         | ログファイルの名前です。エクスポート時に生成されたログファイルもController上にコピーしてある場合は、異なるログファイル名を指定してください。<br><br>デフォルト：「 <a href="#">ログとエラー処理</a> 」を参照してください。                                                                           |
| -NoLog                          | ログファイルを生成しません。このパラメーターを指定すると、-LogFileパラメーターが無視されます。<br><br>デフォルト：False。ログファイルが生成されます。                                                                                                                     |
| -NoClobber                      | -LogFileパラメーターで指定した既存のログファイルの上書きを禁止します。指定したログファイルが存在しない場合、このパラメーターは無視されます。<br><br>デフォルト：False。既存のファイルが上書きされます。                                                                                            |
| -NoDetails                      | スクリプト実行の詳細なレポートをコンソールに表示しません。<br><br>デフォルト：False。詳細なレポートがコンソールに表示されます。                                                                                                                                    |
| -<br>SuppressLogo               | メッセージ「XenApp 6.x to XenApp/XenDesktop 7.6 Migration Tool Version #<yyyyMMdd-hhmm>#」をコンソールに表示しません。トラブルシューティング時にスクリプトのバージョンを確認する場合に便利のため、このパラメーターを指定することは推奨されません。<br><br>デフォルト：False。上記のメッセージがコンソールに表示されます。 |
| -Preview                        | プレビューインポートを実行します。この処理では、入力XMLファイルからデータが読み込まれますが、サイトへのオブジェクトのインポートは実行されません。プレビューインポートにより、実際のインポートで実行される処理内容がログファイルおよびコンソール                                                                                 |

|        |                                       |
|--------|---------------------------------------|
| パラメーター | に出力されます。これにより、管理者はインポートの問題を事前に解決できます。 |
|        | 説明<br>デフォルト：False。実際のインポートが実行されます。    |

例：次のコマンドレットを実行すると、MyPolicies.xmlという名前のXMLファイルからポリシーデータがインポートされます。この処理は、MyPolicies.logという名前のログファイルに記録されます。

```
Import-Policy -XmlInputFile ".\MyPolicies.XML" -LogFile ".\MyPolicies.Log"
```

9. Import-XAFarmコマンドレットを実行して、アプリケーションデータをインポートします。

| パラメーター                        | 説明                                                                                                                                                                                                        |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -XmlInputFile "<string>.xml"  | 入力するXMLファイルの名前です。Export-XAFarmコマンドレットで収集されたデータが記述されています。拡張子として「.xml」を指定する必要があります。<br><br>デフォルト：なし。必須パラメーターです。                                                                                             |
| -XsdFile "<string>"           | XSDファイルの名前です。インポートスクリプトでは、このファイルにより入力XMLファイルの構文が検証されます。使用方法については、「 <a href="#">高度な使用方法</a> 」を参照してください。<br><br>デフォルト：XAFarmData.XSD。                                                                       |
| -LogFile "<string>"           | ログファイルの名前です。エクスポート時に生成されたログファイルもController上にコピーしてある場合は、異なるログファイル名を指定してください。<br><br>デフォルト：「 <a href="#">ログとエラー処理</a> 」を参照してください。                                                                           |
| -NoLog                        | ログファイルを生成しません。このパラメーターを指定すると、-LogFileパラメーターが無視されます。<br><br>デフォルト：False。ログファイルが生成されます。                                                                                                                     |
| -NoClobber                    | -LogFileパラメーターで指定した既存のログファイルの上書きを禁止します。指定したログファイルが存在しない場合、このパラメーターは無視されます。<br><br>デフォルト：False。既存のファイルが上書きされます。                                                                                            |
| -NoDetails                    | スクリプト実行の詳細なレポートをコンソールに表示しません。<br><br>デフォルト：False。詳細なレポートがコンソールに表示されます。                                                                                                                                    |
| -SuppressLogo                 | メッセージ「XenApp 6.x to XenApp/XenDesktop 7.6 Migration Tool Version #<yyyyMMdd-hhmm>#」をコンソールに表示しません。トラブルシューティング時にスクリプトのバージョンを確認する場合に便利なため、このパラメーターを指定することは推奨されません。<br><br>デフォルト：False。上記のメッセージがコンソールに表示されます。 |
| -Preview                      | プレビューインポートを実行します。この処理では、入力XMLファイルからデータが読み込まれますが、サイトへのオブジェクトのインポートは実行されません。プレビューインポートにより、実際のインポートで実行される処理内容がログファイルおよびコンソールに出力されます。これにより、管理者はインポートの問題を事前に解決できます。<br><br>デフォルト：False。実際のインポートが実行されます。        |
| -DeliveryGroupName "<string>" | インポートされるすべてのアプリケーションのデリバリーグループです。使用方法については、 <a href="#">高度な使用方法</a> 」を参照してください。<br><br>デフォルト："<xenapp-farm-name> - Delivery Group"                                                                        |
| -MatchFolder "<string>"       | にマッチする名前のフォルダー内のアプリケーションのみをインポートします。使用方法については、 <a href="#">高度な使用方法</a> 」を参照してください。<br><br>デフォルト：の指定なし。                                                                                                    |

|                                          |                                                                                                                                       |
|------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| パラメーター<br>-NotMatchFolder "<br><string>" | 説明<br>にマッチしない名前のフォルダー内のアプリケーションのみをインポートします。使用方法については、 <a href="#">高度な使用方法</a> を参照してください。<br><br>デフォルト：の指定なし。                          |
| -MatchServer "<string>"                  | にマッチする名前のサーバーからのアプリケーションのみをインポートします。使用方法については、 <a href="#">高度な使用方法</a> を参照してください。                                                     |
| -NotMatchServer "<br><string>"           | にマッチしない名前のサーバーからのアプリケーションのみをインポートします。使用方法については、 <a href="#">高度な使用方法</a> を参照してください。<br><br>デフォルト：の指定なし。                                |
| -MatchWorkerGroup "<br><string>"         | にマッチする名前のワーカーグループに公開されたアプリケーションのみをインポートします。使用方法については、 <a href="#">高度な使用方法</a> を参照してください。<br><br>デフォルト：の指定なし。                          |
| -<br>NotMatchWorkerGroup<br>"<string>"   | にマッチしない名前のワーカーグループに公開されたアプリケーションのみをインポートします。使用方法については、 <a href="#">高度な使用方法</a> を参照してください。<br><br>デフォルト：の指定なし。                         |
| -MatchAccount "<br><string>"             | にマッチする名前のユーザーアカウントに公開されたアプリケーションのみをインポートします。使用方法については、 <a href="#">高度な使用方法</a> を参照してください。<br><br>デフォルト：の指定なし。                         |
| -NotMatchAccount "<br><string>"          | にマッチしない名前のユーザーアカウントに公開されたアプリケーションのみをインポートします。使用方法については、 <a href="#">高度な使用方法</a> を参照してください。<br><br>デフォルト：の指定なし。                        |
| -IncludeStreamedApps                     | 種類が「StreamedToClientOrServerInstalled」のアプリケーションのみをインポートします。ほかの種類のストリーム配信アプリケーションはインポートされません。<br><br>デフォルト：ストリーム配信アプリケーションはインポートされません。 |
| -IncludeDisabledApps                     | 無効に設定されたアプリケーションをインポートします。<br><br>デフォルト：無効に設定されたアプリケーションはインポートされません。                                                                  |

例：次のコマンドレットを実行すると、MyFarm.xmlという名前のXMLファイルからアプリケーションデータがインポートされます。この処理は、MyFarm.logという名前のログファイルに記録されます。

```
Import-XAFarm -XmlInputFile ".\MyFarm.XML" -LogFile ".\MyFarm.Log"
```

10. インポート処理が正しく完了したら、移行後のタスクを行います。

#### 移行後のタスク

XenApp 6.xのポリシーおよびファール設定をXenApp 7.6サイトにインポートしたら、以下のタスクを行ってインポート内容を確認します。

##### ● ポリシーおよびポリシーの設定項目

ポリシーのインポートでは、ポリシー情報が単にコピーされます。ただし、廃止予定のポリシーや設定項目はコピーされません。移行後の検証タスクは、**質的に移行前後の比較**になります。

1. ログファイルには、インポートされたり無視されたりしたすべてのポリシーおよび設定項目が記録されます。まず、ログファイルを参照してインポートされなかったポリシーおよび設定項目を確認します。
2. XenApp 6.xのポリシーとXenApp 7.6にインポートされたポリシーを比較します。同じ設定値が保持されていることを確認します（次の手順で説明する廃止予定の設定項目を除く）。

- 移行したポリシーの数が多い場合は、XenApp 6.xのAppCenterとXenApp 7.6のStudioの両方でポリシーを並べて表示して視覚的に比較できます。
- ポリシーの数が多い場合、視覚的に比較することは困難です。この場合、ポリシーのエクスポートコマンドレット (Export-Policy) を使用してXenApp 7.6のポリシーを別名のXMLファイルにエクスポートして、テキストファイルの比較ツール (windiffなど) を使用してXenApp 6.xからエクスポートしたファイルと比較します。

3. 「**インポートされないポリシー設定**」を参照して、インポート時に変更された設定項目を確認します。XenApp 6.xのポリシーに廃止予定の設定項目のみが含まれる場合、そのポリシーはインポートされません。たとえば、XenApp 6.xで [サーバーヘルス監視テスト] 設定のみが構成されたポリシーがある場合、対応する設定項目がXenApp 7.6にないため、そのポリシーは無視されます。

一部のXenApp 6.xポリシー設定はサポートされなくなりましたが、同等の機能がXenApp 7.6に実装されています。たとえば、XenApp 7.6では、デリバリーグループを編集してサーバーOSマシンの再起動スケジュールを構成できます。この機能は、以前はポリシー設定で実装されていました。

4. XenApp 7.6サイトとXenApp 6.xファームでのポリシーの適用フィルターの動作について確認します。XenApp 6.xファームとXenApp 7.6サイトとのフィルターの違いについては、次のセクションを参照してください。

● **フィルター**

各ポリシーに適用されるフィルター (割り当て先) を慎重に確認してください。XenApp 6.xでの元のフィルター設定がXenApp 7.6でも正しく適用するには、設定の変更が必要になる場合があります。

| フィルター              | 注意事項                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| アクセス制御             | XenApp 6.xからインポートしたアクセス制御フィルターには元の設定と同じ値が適用されており、変更は不要です。                                                                                                                                                                                                                                                                                                                                                                                    |
| Citrix CloudBridge | 単純なブール値なので変更は不要です。                                                                                                                                                                                                                                                                                                                                                                                                                           |
| クライアントのIPアドレス      | 接続が許可または拒否されるクライアントIPアドレスの範囲の一覧です。インポートしたフィルターには元の設定と同じ値が適用されていますが、XenApp 7.6のVDAマシンに接続するクライアントが異なる場合は変更が必要です。                                                                                                                                                                                                                                                                                                                               |
| クライアント名            | クライアントのIPアドレスと同様、インポートしたフィルターには元の設定と同じ値が適用されていますが、XenApp 7.6のVDAマシンに接続するクライアントが異なる場合は変更が必要です。                                                                                                                                                                                                                                                                                                                                                |
| 組織単位               | インポート時に組織単位の解決が可能な場合は元の設定と同じ値が保持されます。特にXenApp 6.xとXenApp 7.6のマシンが異なるドメインに属している場合は、このフィルターの内容を確認してください。適切な値が設定されていない場合、ポリシーが不正な組織単位に適用されてしまいます。<br><br>組織単位は名前でのみ解決されます。このため、XenApp 6.xドメインの組織単位とは異なるメンバーシップを持つ組織単位に解決される可能性もあります。組織単位フィルターのいくつかの値が保持されている場合も、すべての値を確認してください。                                                                                                                                                                 |
| ユーザーまたはグループ        | インポート時にアカウントの解決が可能な場合は元の設定と同じ値が保持されます。<br><br>組織単位フィルターと同様、アカウントは名前でのみ解決されます。このため、XenApp 7.6サイトに同名のドメインおよびユーザーが存在する場合、XenApp 6.xドメインのユーザーとは異なるユーザーに解決される場合があります。フィルターの値を確認して適切に変更しないと、ポリシーが不正なアカウントに適用されてしまいます。                                                                                                                                                                                                                              |
| ワーカーグループ           | XenApp 7.6では、ワーカーグループはサポートされません。XenApp 7.6サイトでは、デリバリーグループ、デリバリーグループの種類、およびタグを使用してポリシーを割り当ててください (これらのフィルターはXenApp 6.xでは使用できません)。 <ul style="list-style-type: none"> <li>● デリバリーグループ: ポリシーを適用するデリバリーグループを指定します。このフィルターでは、既存のデリバリーグループを選択して許可または禁止を指定します。</li> <li>● デリバリーグループの種類: ポリシーを適用するデリバリーグループの種類を指定します。このフィルターでは、デリバリーグループの種類を選択して許可または禁止を指定します。</li> <li>● タグ: ポリシーを適用するVDAマシンのタグを指定します。このフィルターでは、既存のタグを選択して許可または禁止を指定します。</li> </ul> |

XenApp 6.xファームとXenApp 7.6サイトのドメインが異なる場合は、特にドメインユーザーの変更による影響を受けるフィルターについて詳細に確認してください。インポートスクリプトでは、ドメインやユーザーの名前が文字列でのみ解決され、移行先ドメインのユーザーが決定されます。このため、一部のアカウントが正しく解決されない場合があります。異なるドメインでドメイン名やユーザー名が同一であることはまれですが、これらのフィルターの値が正しく移行されていることを確認してください。

## ● アプリケーション

アプリケーションをインポートするスクリプトでは、アプリケーションだけではなく、デリバリーグループなどのオブジェクトも作成されます。アプリケーションを複数回に分けてインポートすると、元のアプリケーションフォルダーの構造が大幅に変更されている場合があります。

1. まず、移行ログファイルを確認して、インポートおよび除外されたアプリケーションと、アプリケーションの作成に使用されたコマンドレットの詳細を確認します。
2. 各アプリケーションについて、以下を確認します。
  - 基本的なプロパティが正しく保持されていることを視覚的に確認します。「[アプリケーションプロパティの対応](#)」を参照して、変更されずにインポートされるプロパティ、インポートから除外されるプロパティ、またはXenApp 6.xのアプリケーションデータで初期化されるプロパティについて確認します。
  - ユーザーの一覧を確認します。インポートスクリプトにより、指定ユーザーの一覧がXenApp 7.6の「表示の制限」のユーザー一覧としてインポートされます。この一覧で各ユーザーが正しく保持されていることを確認してください。
3. アプリケーションサーバーはインポートされません。このため、インポートした直後のアプリケーションにユーザーがアクセスすることはできません。ユーザーがアクセスできるようにするには、これらのアプリケーションを含んでいるデリバリーグループを、アプリケーションの実行可能イメージを含んでいるマシンカタログに割り当てる必要があります。各アプリケーションについて、以下を確認します。
  - 実行可能ファイルの名前と作業ディレクトリが、そのデリバリーグループのマシンカタログで割り当てられているマシン上に存在すること。
  - コマンドラインパラメーター（ファイル名、環境変数、または実行可能ファイル名など）。そのデリバリーグループのマシンカタログで割り当てられているすべてのマシン上で、それらのパラメーターが有効である必要があります。

## ● ログファイル

ログファイルには、エクスポートおよびインポートの処理に関する重要な情報が記録されています。既存のログファイルがデフォルトで上書きされず、固のファイル名で生成される仕様になっているのはこのためです。

前の「ログとエラー処理」で説明したように、PowerShellのStart-TranscriptおよびStop-Transcriptコマンドレットを使って入出力のすべてを記録した場合は、その記録とログファイルを参照することでエクスポートおよびインポートの詳細を確認することができます。

ログファイルに記録されるタイムスタンプは、特定の問題について診断するときに便利です。たとえば、エクスポートまたはインポートに時間がかかった場合は、データベース接続に問題があったかどうか、またはユーザーアカウントの解決に時間がかかったかどうかを確認できます。

ログファイルに記録されたコマンドにより、移行するオブジェクトがどのように読み取られ、作成されたかを判断できます。たとえば、デリバリーグループを作成するときにはいくつかのコマンドが実行され、デリバリーグループオブジェクト自体だけでなく、そのデリバリーグループにアプリケーションオブジェクトを割り当てるためのアクセスポリシー規則などのオブジェクトも作成されます。

エクスポートやインポートに失敗した場合もログファイルが役に立ちます。通常、ログファイルの最後の行を参照すると問題の原因が判断できる場合があります。ログファイルには、エラーメッセージも記録されます。XMLファイルとログファイルを参照して、問題発生時に処理されていたオブジェクトを特定できます。

移行内容を確認してテストした後で、以下のタスクを行います。

1. XenApp 6.5のワーカーサーバーを最新のVirtual Delivery Agent (VDA) にアップグレードします。これを行うには、ワーカーサーバー上でXenApp 7.6のインストーラーを実行します。これにより、XenApp 6.5がアンインストールされ、最新のVDAがインストールされます。手順については、「[XenApp 6.5ワーカーのVDA for Windows Server OSへのアップグレード](#)」を参照してください。  
XenApp 6.0のワーカーサーバーでは、手作業でXenApp 6.0をアンインストールします。その後でXenApp 7.6のインストーラーを使用して最新のVDAをインストールします。XenApp 7.6のインストーラーを実行してXenApp 6.0をアンインストールすることはできません。
2. 新しいXenAppサイトのStudioを使用して、そのワーカーのマシンカタログを作成（または既存のマシンカタログを編集）します。
3. そのマシンカタログを使用してデリバリーグループを作成し、アップグレードしたマシンおよびそれらのマシン上のアプリケーションをユーザーに提供します。

## 高度な使用方法

デフォルトでは、Export-PolicyコマンドレットによりすべてのポリシーデータがXMLファイルにエクスポートされます。同様に、Export-XAFarmコマンドレットによりすべてのファームデータがXMLファイルにエクスポートされます。ここでは、コマンドラインパラメーターを使用してエクスポートおよびインポートするオブジェクトを制御する方法について説明します。

- **一部のアプリケーションのみのエクスポート** - 移行するアプリケーションの数が多き場合は、1つのXMLファイルにエクスポートするアプリケーションの数を制御できます。これを行うには、以下のパラメーターを使用します。
  - AppLimit - エクスポートするアプリケーションの数を指定します。
  - SkipApps - 指定した数のアプリケーションのエクスポートが省略され、その数以降のアプリケーションからエクスポートされます。これらのパラメーターを使用して、アプリケーションをいくつかのグループに分けてエクスポートできます。たとえば、1回目のExport-XAFarmの実行時に最初の200個のアプリケーションをエクスポートするには、AppLimitパラメーターを次のように指定します。  
Export-XAFarm -XmlOutputFile "Apps1-200.xml" -AppLimit "200"  
2回目のExport-XAFarmの実行時に残りのアプリケーションのうちの100個をエクスポートするには、エクスポート済みの200個のアプリケーションをエクスポートから除外するためのSkipAppsパラメーターと、その後の100個をエクスポートするためのAppLimitパラメーターを次のように指定します。  
Export-XAFarm -XmlOutputFile "Apps201-300.xml" -AppLimit "100" -SkipApps "200"
- **エクスポートから除外するオブジェクトの指定** - インポートされないオブジェクトなど、一部のオブジェクトはエクスポート時に無視されます。これらのオブジェクトについては、「[インポートされないポリシー設定](#)」および「[アプリケーションプロパティの対応](#)」を参照してください。以下のパラメーターを使

用すると、特定の種類のオブジェクトをエクスポートから除外できます。

- IgnoreAdmins - 管理者オブジェクトをエクスポートから除外します。
- IgnoreServers - サーバーオブジェクトをエクスポートから除外します。
- IgnoreZones - ゾーンオブジェクトをエクスポートから除外します。
- IgnoreOthers - そのほかのオブジェクト (構成ログ、負荷評価基準、負荷分散ポリシー、プリンタードライバー、ワーカークラウド) をエクスポートから除外します。
- IgnoreApps - アプリケーションオブジェクトをエクスポートから除外します。アプリケーション以外のデータとアプリケーションを別のXMLファイルにエクスポートする場合に使用します。

これらのパラメーターは、エクスポート時の問題を回避するときにも使用することもできます。たとえば、ゾーン内に不適切なサーバーがあるとエクスポートが失敗する場合があります。この場合は、IgnoreZonesパラメーターを使用してほかのオブジェクトをエクスポートできます。

- **デリバリーグループ名** - インポートするアプリケーションを複数のデリバリーグループ内に分けて追加する場合 (異なるユーザーグループに提供する場合や異なるサーバーグループで公開する場合など)、Import-XAFarmコマンドレットを複数に分けて実行し、異なるアプリケーションとデリバリーグループを指します。移行後にPowerShellコマンドレットを使ってアプリケーションをほかのデリバリーグループに移動することもできますが、インポート時にデリバリーグループを指定しておくと、後でアプリケーションを移動する手間が省けます。

1. DeliveryGroupNameパラメーターを指定してImport-XAFarmコマンドレットを実行します。これにより、指定されたデリバリーグループが存在しない場合は作成されます。

2. 以下のパラメーターで正規表現を使用して、アプリケーションを特定のフォルダー、ワーカークラウド、ユーザーアカウント、サーバー名に基づいて特定のデリバリーグループにインポートします。正規表現の部分を引用符または二重引用符で囲むことをお勧めします。正規表現について詳しくは、[http://msdn.microsoft.com/ja-jp/library/hs600312\(v=vs.110\).aspx](http://msdn.microsoft.com/ja-jp/library/hs600312(v=vs.110).aspx)を参照してください。

- MatchWorkerGroupおよびNotMatchWorkerGroup - たとえば、ワーカークラウドに公開されたアプリケーションに対して次のコマンドレットを実行すると、ワーカークラウド「Productivity Apps」内のアプリケーションがXenApp 7.6の同じ名前のデリバリーグループにインポートされます。

```
Import-XAFarm -XmlInputFile XAFarm.xml -LogFile XAFarmImport.log -MatchWorkerGroup 'Productivity Apps' -DeliveryGroupName 'Productivity Apps'
```

- MatchFolderおよびNotMatchFolder - たとえば、アプリケーションフォルダーで分類されたアプリケーションに対して次のコマンドレットを実行すると、フォルダー「Productivity Apps」内のアプリケーションがXenApp 7.6の同じ名前のデリバリーグループにインポートされます。

```
Import-XAFarm -XmlInputFile XAFarm.xml -LogFile XAFarmImport.log -MatchFolder 'Productivity Apps' -DeliveryGroupName 'Productivity Apps'
```

たとえば、次のコマンドレットを実行すると、「MS Office Apps」という文字列を含んでいる名前のフォルダーからすべてのアプリケーションがデフォルトのデリバリーグループにインポートされます。

```
Import-XAFarm -XmlInputFile .\TheFarmApps.XML -MatchFolder '.*MS Office Apps.*'
```

- MatchAccountおよびNotMatchAccount - たとえば、Active Directoryのユーザーまたはユーザーグループに公開されたアプリケーションに対して次のコマンドレットを実行すると、ユーザーグループ「Finance Group」に公開されたアプリケーションがXenApp 7.6のデリバリーグループ「Finance」にインポートされます。

```
Import-XAFarm -XmlInputFile XAFarm.xml -LogFile XAFarmImport.log -MatchAccount 'DOMAIN\Finance Group' -DeliveryGroupName 'Finance'
```

- MatchServerおよびNotMatchServer - たとえば、サーバー単位で公開されたアプリケーションに対して次のコマンドを実行すると、名前が「Current」でないサーバーに関連付けられたアプリケーションが「Legacy」という名前のXenAppデリバリーグループにインポートされます。

```
Import-XAFarm -XmlInputFile XAFarm.xml -LogFile XAFarmImport.log -NotMatchServer 'Current' -DeliveryGroupName 'Legacy'
```

- **カスタマイズ** - PowerShellのプログラミング知識がある場合は、カスタムのツールを作成できます。たとえば、エクスポートスクリプトをインベントリツールとして使用して、XenApp 6.xファームの変更履歴を追跡できます。また、XSDファイルを編集したり新しく作成したりして、XMLファイルに記述されるデータを追加したり、記述形式を変更したりできます。各インポートコマンドレットに非デフォルトのXSDファイルを指定できます。

注: 特定の移行要件に応じてスクリプトファイルを編集することができますが、その場合はサポート対象外になります。Citrixのテクニカルサポートは、スクリプトを元の状態に戻したうえで正しい動作を確認することを推奨し、必要な場合サポートを提供します。

## トラブルシューティング

- PowerShell Verison 2.0を使用する環境で、Add-PSSnapInコマンドレットを使用してCitrix Group Policy PowerShellプロバイダースナップインまたはCitrix Common Commandsスナップインを追加した場合、エクスポートまたはインポートコマンドレットの実行時に「オブジェクト参照がオブジェクトインスタンスに設定されていません」というエラーメッセージが表示されることがあります。このエラーメッセージはスクリプトの実行に影響せず、無視して構いません。

- Citrix Group Policy PowerShellプロバイダースナップインはエクスポートおよびインポートのスクリプトモジュールにより自動的に追加されるため、これらのスクリプトモジュールを使用するコンソールセッション上にこのスナップインを追加したり削除したりすることは避けてください。このスナップインを前に追加したり削除したりすると、以下のエラーメッセージが表示されることがあります。

- 「名前 'LocalGpo' のドライブは既に存在しています。」このエラーメッセージは、スナップインを2回追加すると表示されます。スナップインのロード時にドライブ「LocalGpo」のマウントが試行されるため、このメッセージが表示されます。

- 「パラメーター名 'Controller' に一致するパラメーターが見つかりません。」このエラーメッセージは、スナップインが追加されていないにもかかわらずドライブのマウントが試行されると表示されます。この問題は、スナップインが事前に削除されていると発生します。コンソールを閉じて新しいセッションを起動してください。新しいセッションでスクリプトモジュールをインポートします。事前にスナップインを追加したり削除したりしないでください。

- モジュールをインポートするときにPSD1ファイルを右クリックして [開く] または [PowerShellで開く] を選択すると、プロセスを終了するまでPowerShellコンソールのウィンドウが高速に開いたり閉じたりします。この問題を避けるには、PowerShellコンソールのウィンドウにPowerShellスクリプトモジュールの名前を直接入力してください (「Import-Module .\ExportPolicy.psd1」など)。

- エクスポートまたはインポートの実行時にアクセス権に関するエラーが発生した場合は、オブジェクトの読み取り権限 (エクスポート時) または読み取り



よび作成権限（インポート）を持つXenApp管理者アカウントを使用していることを確認してください。また、PowerShellスクリプトを実行するためのWindows権限も必要です。

- エクスポートに失敗した場合は、XenApp 6.xファームのヘルス状態が良好であることを確認してください。これを行うには、XenApp 6.xのコントローラーサーバー上でDSMAINTおよびDSCHECKコマンドを使用します。
- プレビューインポートを実行した後で実際のインポートを実行しても何もインポートされない場合は、インポートコマンドレットから-Previewパラメータを削除したことを確認してください。

#### インポートされないポリシー設定

以下のコンピューターポリシー設定とユーザーポリシー設定は、サポートされていないためインポートされません。フィルターを適用していないポリシーはインポートされないことに注意してください。これらの設定項目をサポートする機能やコンポーネントが新しいテクノロジーやコンポーネントに置き換えられれば、アーキテクチャやプラットフォームの変更によって適用されなくなりました。

#### インポートされないコンピューターポリシー設定

- 接続のアクセス制御
- サーバーのCPU最適化レベル
- DNSアドレス解決
- ファーム名
- アイコンの完全キャッシュ
- サーバーヘルス監視、サーバーヘルス監視テスト
- ライセンスサーバーのホスト名、ライセンスサーバーポート
- ユーザーセッションの制限、管理者アカウントの接続も制限する
- 負荷評価基準名
- ログオン数制限のログ
- ログオン制御できる最大サーバー数 (%)
- メモリの最適化、メモリ最適化のアプリケーション除外一覧、メモリの最適化の間隔、メモリの最適化のスケジュール：月間、メモリの最適化のスケジュール：週間、メモリの最適化のスケジュール：時間
- オフラインアプリケーションへの再認証なしでの再接続、オフラインアプリケーションのログ、オフラインライセンスの有効期間、オフラインアプリケーションユーザー
- パスワードを要求する
- カスタムの再起動警告、カスタムの再起動警告メッセージ、再起動前のログオンの無効化、再起動スケジュールの頻度、再起動スケジュールのランダム化間隔、再起動スケジュールの開始日、再起動スケジュールの時間、再起動警告の間隔、再起動警告の開始時間、ユーザーへの再起動警告、スケジュールによる再起動
- シャドウ機能 \*
- XML要求を信頼する (StoreFrontで構成)
- 仮想IPアダプターアドレスフィルター、仮想IP互換プログラム一覧、仮想IP拡張互換性、仮想IPアダプターアドレスフィルタープログラム一覧
- ワークロード名
- XenApp製品エディション、XenApp製品モデル
- XML Serviceのポート

\* Windowsリモートアシスタンス機能に置き換えられています

#### インポートされないユーザーポリシー設定

- クライアントCOMポートを自動接続する、クライアントLPTポートを自動接続する
- クライアントCOMポートリダイレクト、クライアントLPTポートリダイレクト
- クライアントプリンター名
- 同時接続数の制限
- シャドウする側からの入力 \*
- 残留セッションの切断タイマー、残留セッションの終了タイマー
- シャドウイベントをログに記録する \*
- シャドウ要求をシャドウされる側のユーザーに通知する \*
- 事前起動セッションの切断タイマー、事前起動セッションの終了タイマー
- セッションの重要度
- Single Sign-On、Single Sign-On中央ストア
- ほかのユーザーをシャドウできるユーザー、ほかのユーザーをシャドウできないユーザー \*

\* Windowsリモートアシスタンス機能に置き換えられています

#### インポートされないアプリケーションの種類

以下の種類のアプリケーションはインポートされません。

- サーバーのデスクトップ
- 公開コンテンツ

- ストリーム配信アプリケーション（アプリケーションのストリーム配信にはApp-Vを使用します）

#### アプリケーションプロパティの対応

ファームデータのインポートスクリプトでは、アプリケーションのみがインポートされます。以下のアプリケーションプロパティは変更されずにインポートされます。

| IMAのプロパティ             | FMAのプロパティ                |
|-----------------------|--------------------------|
| AddToClientDesktop    | ShortcutAddedToDesktop   |
| AddToClientStartMenu  | ShortcutAddedToStartMenu |
| ClientFolder          | ClientFolder             |
| CommandLineExecutable | CommandLineExecutable    |
| CpuPriorityLevel      | CpuPriorityLevel         |
| 説明                    | 説明                       |
| DisplayName           | PublishedName            |
| 有効                    | 有効                       |
| StartMenuFolder       | StartMenuFolder          |
| WaitOnPrinterCreation | WaitForPrinterCreation   |
| WorkingDirectory      | WorkingDirectory         |
| FolderPath            | AdminFolderName          |

注：IMAとFMAでは、フォルダー名の長さの制限が異なります。IMAでのフォルダー名は256文字以下、FMAでは64文字以下である必要があります。フォルダーパスに64文字を超えるフォルダー名が含まれる場合、そのフォルダー内のアプリケーションはインポートされません。この制限はフォルダーパスに含まれるフォルダー名に対するもので、フォルダーパス全体に対するものではありません。アプリケーションが正しくインポートされるように、アプリケーションのフォルダー名の長さを確認し、必要な場合は短くしてからエクスポートしてください。

以下のアプリケーションプロパティには、デフォルトで初期化されるものと初期化されないものがあります。また、XenApp 6.xのデータで指定された値が設定されるものもあります。

| FMAのプロパティ                     | 値                                                                                        |
|-------------------------------|------------------------------------------------------------------------------------------|
| 名前                            | フルパス名に初期化されます。フルパスには、IMAプロパティのFolderPathおよびDisplayNameが含まれ、先頭の「Applications\」の部分は削除されます。 |
| ApplicationType               | HostedOnDesktop                                                                          |
| CommandLineArguments          | XenApp 6.xのコマンドライン引数で初期化されます。                                                            |
| IconFromClient                | 初期化されずにデフォルト値のfalseになります。                                                                |
| IconUid                       | XenApp 6.xアイコンデータで作成されたアイコンオブジェクトに初期化されます。                                               |
| SecureCmdLineArgumentsEnabled | 初期化されずにデフォルト値のtrueになります。                                                                 |
| UserFilterEnabled             | 初期化されずにデフォルト値のfalseになります。                                                                |

| FMAのプロパティ | 値                             |
|-----------|-------------------------------|
| UUID      | 読み取り専用でControllerにより割り当てられます。 |
| Visible   | 初期化されずにデフォルト値のtrueになります。      |

以下のアプリケーションプロパティは、一部のみ移行されます。

| IMAのプロパティ | コメント                                                                                                                                                             |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FileTypes | 新しいXenAppサイトに存在するファイルタイプのみが移行されます。新しいサイトに存在しないファイルタイプは無視されます。ファイルタイプは、新しいサイトのファイルタイプが更新された後でインポート可能になります。                                                        |
| IconData  | エクスポートされたアプリケーションのアイコンデータが提供されている場合は新しいアイコンオブジェクトが作成されます。                                                                                                        |
| Accounts  | アプリケーションのユーザーアカウントはデリバリーグループとアプリケーションのユーザー一覧に分類されます。指定ユーザーのアカウントはアプリケーションのユーザー一覧を初期化するために使用されます。さらに、ユーザーアカウントが属するドメインのDomain Usersアカウントがデリバリーグループのユーザー一覧に追加されます。 |

以下のXenApp 6.xプロパティはインポートされません。

| IMAのプロパティ                              | コメント                                         |
|----------------------------------------|----------------------------------------------|
| ApplicationType                        | 無視されます。                                      |
| HideWhenDisabled                       | 無視されます。                                      |
| AccessSessionConditions                | デリバリーグループのアクセスポリシーに置き換えられます。                 |
| AccessSessionConditionsEnabled         | デリバリーグループのアクセスポリシーに置き換えられます。                 |
| ConnectionsThroughAccessGatewayAllowed | デリバリーグループのアクセスポリシーに置き換えられます。                 |
| OtherConnectionsAllowed                | デリバリーグループのアクセスポリシーに置き換えられます。                 |
| AlternateProfiles                      | FMAはストリーム配信アプリケーションをサポートしません。                |
| OfflineAccessAllowed                   | FMAはストリーム配信アプリケーションをサポートしません。                |
| ProfileLocation                        | FMAはストリーム配信アプリケーションをサポートしません。                |
| ProfileProgramArguments                | FMAはストリーム配信アプリケーションをサポートしません。                |
| ProfileProgramName                     | FMAはストリーム配信アプリケーションをサポートしません。                |
| RunAsLeastPrivilegedUser               | FMAはストリーム配信アプリケーションをサポートしません。                |
| AnonymousConnectionsAllowed            | 認証が不要なユーザー（匿名ユーザー）の接続は、FMAでは別の方法でサポートされています。 |
| ApplicationId、SequenceNumber           | IMA固有のデータです。                                 |
| AudioType                              | FMAはクライアント接続の詳細オプションをサポートしません。               |
| EncryptionLevel                        | SecureICAの有効/無効はデリバリーグループで設定します。             |

| IMAのプロパティ                            | コメント                              |
|--------------------------------------|-----------------------------------|
| EncryptionRequired                   | SecureICAの有効/無効はデリバリーグループで設定します。  |
| SslConnectionEnabled                 | FMAでは別の方法でSSLが実装されています。           |
| ContentAddress                       | FMAは公開コンテンツをサポートしません。             |
| ColorDepth                           | FMAはウィンドウ表示の詳細オプションをサポートしません。     |
| MaximizedOnStartup                   | FMAはウィンドウ表示の詳細オプションをサポートしません。     |
| TitleBarHidden                       | FMAはウィンドウ表示の詳細オプションをサポートしません。     |
| WindowsType                          | FMAはウィンドウ表示の詳細オプションをサポートしません。     |
| InstanceLimit                        | FMAはアプリケーション数の制限をサポートしません。        |
| MultipleInstancesPerUserAllowed      | FMAはアプリケーション数の制限をサポートしません。        |
| LoadBalancingApplicationCheckEnabled | 負荷分散は、FMAでは別の方法でサポートされています。       |
| PreLaunch                            | セッションの事前起動は、FMAでは別の方法でサポートされています。 |
| CachingOption                        | セッションの事前起動は、FMAでは別の方法でサポートされています。 |
| ServerNames                          | FMAでは別の方法が使用されています。               |
| WorkerGroupNames                     | FMAはワーカーグループをサポートしません。            |

# XenDesktop 4からの移行

Jan 04, 2017

移行ツールを使用して、XenDesktop 4ファームのデータや設定をXenDesktop 7.xサイトに転送できます。移行ツールは、XenDesktopインストールメディアの [Support] > [Tools] > [MigrationTool] フォルダーにあります。このツールには、以下のコンポーネントが含まれます。

- エクスポートツールXdExport。このツールにより、XenDesktop 4ファームのデータがXMLファイル（デフォルト名：XdSettings.xml）にエクスポートされます。XMLファイルスキーマはファイルXdFarm.xsd内にあります。
- インポートツールXdImport。このツールにより、データをインポートするPowerShellスクリプトImport-XdSettings.ps1が実行されます。

移行ツールを正しく使用するには、XenDesktop 4ファームとXenDesktop 7.xサイトでハイパーバイザーのバージョン（XenServer 6.2など）とActive Directory環境が同じである必要があります。

このツールでXenAppファームを移行することはできません。また、XenDesktop 4ファームをXenAppに移行することはできません。

ヒント：XenDesktop 5（またはそれ以降のバージョンのXenDesktop）は、新しいXenDesktopバージョンにアップグレードできます。詳しくは、「[展開のアップグレード](#)」を参照してください。

## 制限事項

すべてのデータや設定がエクスポートされるわけではありません。以下の構成項目はエクスポートされますがインポートできません。このため、これらの設定は移行できません。

- 管理者
- 委任管理設定
- デスクトップグループフォルダー
- ライセンス構成
- レジストリキー

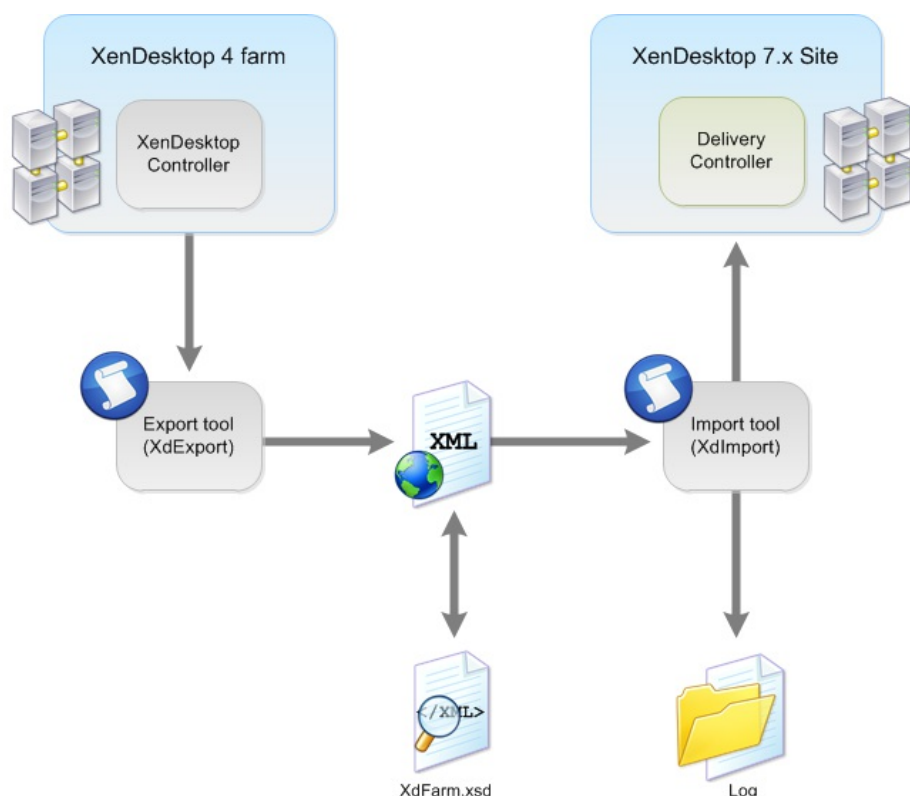
移行時に以下を行うことはできません。

- ポリシー、デスクトップグループ、およびホスト設定のマージ（統合）。
- プライベートデスクトップのランダムデリバリーグループへのマージ。
- 移行ツールによる既存のコンポーネント設定の変更。

詳しくは、「[移行される項目と移行されない項目](#)」を参照してください。

## 移行手順

次の図は、移行プロセスの概要です。



以下の順序で移行します。

1. XenDesktop 4 Controller上のStudioコンソールで、エクスポートするすべてのマシンをメンテナンスモードにします。
2. XdExportを使って、XenDesktop 4ファームからXMLファイルにデータおよび設定をエクスポートします。 [XenDesktop 4ファームからのエクスポート](#) を参照してください。
3. エクスポートされたXMLファイルを編集して、新しいXenDesktopサイトにインポートしないデータや設定を削除して、必要な変更を行います。「[移行ツールXMLファイルの編集](#)」を参照してください。
4. XdImportを使って、XMLファイルから新しいXenDesktopサイトにデータおよび設定をインポートします。 [XenDesktop 4データのインポート](#) を参照してください。
5. 変更を追加するには、手順3.と手順4.を繰り返します。変更を追加した後で、既存のデリバリーグループに追加のデスクトップをインポートすることができます。これを行うには、インポートするときにMergedesktopsパラメーターを使用します。
6. 移行後のタスクを実行します。「[移行後のタスク](#)」を参照してください。

## 準備

移行を開始する前に以下のタスクを完了しておく必要があります。

- エクスポートおよびインポートできるデータの種類を確認し、どのように展開に適用できるのかについて把握する必要があります。「[移行される項目と移行されない項目](#)」を参照してください。
- サイトのデータベースを手動でバックアップしておくことを強く推奨します。これにより、問題が発生した場合に元の状態に復元することができます。
- XenDesktop 7.xコンポーネントをインストールして、データベースおよびサイトを作成します。
- XenDesktop 4から移行するには、すべてのVDAをXenDesktop 5.xバージョンにしておく必要があります。これにより、XenDesktop 4とXenDesktop 7.xの両方のControllerとの互換性が保持されます。XenDesktop 7.xへの移行が完全に完了したら、Windows 7上のVDAをXenDesktop 7.xバージョンにアップグレードできます。詳しくは、「[移行例](#)」を参照してください。

# XenDesktop 4ファームからのエクスポート

Jan 04, 2017

エクスポートツールのXdExportは、単一のXenDesktop 4ファームからデータを抽出し、そのデータ値を使ってXMLファイルを作成します。

XMLファイルのスキーマは、移行ツール (XdExport.zipおよびXdImport.zip) と一緒にダウンロードされるXdFarmxsdファイルに定義されています。

移行元のXenDesktop 4ファームで動作するController上でXdExportを実行します。このマシンには、XenDesktop 4 PowerShell SDKがインストールされている必要があります。データをエクスポートするには、以下の権限が必要です。

- ファームの読み取り専用Citrix管理者以上の権限。
- レジストリの読み取り権限。

必要な場合は、ユーザーがVDAにログオンしている場合など、稼働しているXenDesktop Controller上でツールを実行することもできます (ただしお勧めしません)。

以下のことを強くお勧めします。

- ツールを実行するXenDesktop 4 Controllerに最新のパブリックHotfixを適用しておいてください。
- エクスポートの実行中は、デスクトップグループの削除など、サイトの構成を変更しないでください。

1. XdExport.zipをダウンロードして、内容をXenDesktop 4のController上に抽出します。
2. コマンドプロンプトでXdExport.exeを実行します。以下のパラメーターを指定できます。

| パラメーター           | 説明                                                                                                                                                                                    |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -Verbose         | 詳細な進捗情報を提供するメッセージを生成します。                                                                                                                                                              |
| -FilePath <path> | XMLファイルのエクスポート先のパスを指定します。デフォルト値は、.\XdSettings.xmlです。                                                                                                                                  |
| -Overwrite       | -FilePathで指定した場所にあるファイルを上書きします。このパラメーターを指定しない場合、エクスポート先に同名のファイルがあると「Error: File already exists.Specify -Overwrite to allow the file to be overwritten.」というエラーメッセージが表示され、ツールの実行に失敗します。 |
| -?または-help       | 各パラメーターについての説明を表示します。データはエクスポートされません。                                                                                                                                                 |

3. ツールの実行に成功すると、「Done」というメッセージが表示されます。-FilePathパラメーターで指定した場所にXdSettings.xmlファイルが出力されます。ツールの実行に失敗した場合は、エラーメッセージが表示されます。

# 移行ツールXMLファイルの編集

Jan 04, 2017

XenDesktop 7.xサイトへの移行を複数回に分けて実行し、一部のユーザー、デリバリーグループ、およびポリシーを他のデータよりも前にインポートする場合は、エクスポートツール (XdExport) により生成されたXMLファイルの内容を確認し、必要に応じて編集してから、データをインポートする必要があります。

ファイルの内容を表示または変更するには、任意のテキストエディターを使用します。Microsoft XML Notepadなど専用のXMLエディターを使用することもできます。

インポートツール (XdImport) でXMLファイルを正しくインポートするためには、そのファイルに特定のXML要素が存在している必要があります。

必要なXMLスキーマは、移行ツールと一緒にダウンロードされるXdFarm.xsdファイルに定義されています。このファイルを編集する場合は、次の点に注意してください。

- minOccurs属性の値が1以上である場合、その要素が1つ以上記述されていなければなりません。
- インポートツールで指定するXMLファイルが無効な場合は、ツールが停止してXMLファイルの問題箇所を示すエラーメッセージが表示されます。

## デスクトップまたはデリバリーグループのサブセットのインポート

特定のデリバリーグループやデスクトップのみをインポートする場合は、要素の内容を編集します。要素は複数の要素を保持でき、各要素には複数の要素を保持できる要素が1つあります。

すべての要素を削除する場合でも、要素は削除しないでください。同様に、すべての要素を削除する場合でも、各要素内の要素を削除しないでください。

特定のマシンやデリバリーグループを移行しない場合は、その要素または要素全体を削除します。たとえば、次のような内容のXMLファイルがあるとして。

```
... ..
```

このファイルを以下のように編集すると、デリバリーグループGroup1がインポートされなくなり、Group2のMACHINE3のみがインポートされます。

```
... ..
```

## 同じ名前のデリバリーグループの管理

XenDesktop 4のデスクトップグループでは、同じ名前のデスクトップグループを異なるフォルダーで管理できました。また、ユーザーにはこのデスクトップグループ名が表示されました。このリリースのXenDesktopでは、デリバリーグループをフォルダーで管理できないため、一意のデリバリーグループ名が必要です。また、ユーザーには、このデリバリーグループID (内部デリバリーグループ名) とは別のものが表示されます。この仕様の違いに対処するため、移行時にデスクトップグループ名の変更が必要になる場合があります。

たとえば、XenDesktop 4ファームでは、2人の異なるユーザーに対して同じ「My Desktop」という名前の2つの異なるデスクトップグループを作成して、これらを個別のフォルダーで管理できました。これらのデリバリーグループをXenDesktop 7.xサイトに移行する場合、XMLファイルでデスクトップグループ名を変更して一意の名前を付ける必要があります。

また、移行先のXenDesktop 7.xサイトに、移行するデスクトップグループと同じ名前のデリバリーグループがある場合も、XenDesktop 4のデスクトップグループ名をXMLファイルで変更しておく必要があります。ただし、移行するデスクトップグループがXenDesktop 7.xのデリバリーグループと同一で、XMLファイルに出力されたすべてのマシンを既存のデリバリーグループに統合する場合は、デスクトップグループ名を変更する必要はありません。その代わりに、インポートツールにMergeDesktopsパラメーターを指定します。たとえば、次のような内容のXMLファイルがあるとして。



... \Sales ... \Finance

次のように、重複した名前を変更します。

... \Sales ... \Finance

## ポリシーのインポートの管理

XMLファイルから特定のポリシーを削除したり、ポリシー名の重複を避けるために一意の名前を指定したりできます。ポリシーをマージすることはできません。

- XMLファイルのポリシーデータはすべてインポートされます。ただし、何らかの問題がある場合にはいずれのポリシーもインポートされません。
- 多くの設定を含んでいる多数のポリシーのインポートには数時間かかることがあります。
- 複数のポリシーを複数回に分けてインポートすると、元の優先度設定が変更されることがあります。インポートするポリシーの相対的な優先度は維持されますが、インポート先サイトの既存のポリシーよりも高い優先度が設定されます。たとえば、優先度1、2、3、および4の4つのポリシーを2回に分けてインポートする場合、優先度3および4のポリシーを先にインポートする必要があります。これは、後からインポートするポリシーの優先度が自動的に高くなるためです。

XenDesktop 7.xサイト内にポリシーのサブセットのみをインポートするには、要素の内容を編集します。要素には、多くの要素が含まれている場合があります。すべての要素を削除する場合でも、要素は削除しないでください。Policy要素全体を削除して、特定のXenDesktop 4ファームポリシーをインポートするのを防ぎます。たとえば、次のような内容のXMLファイルがあるとします。

... ..

XenDesktop 7.xサイトの既存のポリシーとの競合を避けるために、XenDesktop 4のすべてのポリシーを移行対象から除外する場合は、次のようにすべての要素を削除します。

または、ファイルを編集して次のようにポリシーが別の名前でもインポートされるようにします。

... ..

# XenDesktop 4データのインポート

Jan 04, 2017

インポートツール (XdlImport) は、エクスポートツールのXdExportによって生成されたXMLファイルに含まれているXenDesktop 4の設定を読み取って、既存のXenDesktop 7.xサイトにそれらの設定を適用します。このインポートツールでは、PowerShellスクリプトImport-XdSettings.ps1を使用します。

エクスポートされたデータの一部のみを適用するには、インポートツールを実行する前にXMLファイルを編集します。たとえば、XenDesktop 7.xサイトに移行する必要がないデスクトップグループやポリシーをXMLファイルから削除しておきます。エレメント全体を空のままにしてインポートツールを実行すると、そのエレメントのすべての項目 (デスクトップグループなど) を削除できません。XMLファイルをインポートするときに、そのファイルの内容がツールにより検証されます。

すべてのXenDesktop 7.x SDKがインストールされている任意のマシン上でXdlImportを実行します。このツールを実行するには、XenDesktopに対して完全な権限を持つ管理者アカウントが必要です。

インポートする前に、データベースを含め、XenDesktop 7.xサイトを正しくセットアップしておく必要があります。ユーザーのテストや全般的なサイト構成を行う前に、XenDesktop 7.xへのインポートを完了しておくことを推奨します。既存の構成にインポート内容を統合する場合は、サイトの非使用時にインポートを実行してください。

1. XenDesktop 7.xサイトを作成します。
2. XdlImport.zipをダウンロードして、このツールを実行するマシン上にファイルを抽出します。
3. PowerShellセッションでImport-XdSettings.ps1を実行します。以下のパラメーターを指定します。

| パラメーター                            | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| - HypervisorConnectionCredentials | 必須です。ハイパーバイザー接続の作成に必要なハイパーバイザーアドレスをPSCredentialインスタンスにマップするPowerShellハッシュテーブルです。さらに、XenDesktop 4ファームの接続先ハイパーバイザーの資格情報を入力します。<br><br>単一のハイパーバイザーの場合は、次のような引数を作成します。<br><br><code>\$credential = Get-Credential \$mappings = @"(\"http://\" = \$credential) .\Import-XdSettings.ps1 -FilePath .\XdSettings.xml -HypervisorConnectionCredentials \$mappir</code><br>ハッシュテーブルで指定されるアドレスは、XMLファイルに出力されたアドレスと正確に一致する必要があります。<br><br>たとえば、XenServerおよびVMwareハイパーバイザーでは、次の引数を作成します。<br><br><code>\$Xencredential = Get-Credential \$VMWcredential = Get-Credential \$mappings = @"(\"http://\" = \$Xencredential,\"http://SDK\" = \$VMWcredential) .\Import-XdSetting:</code> |
| -FilePath <path>                  | (の値は必須) ファームデータをインポートするXMLファイルの場所です。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| -AdminAddress                     | XenDesktop 7.xサイトのControllerの名前です。デフォルト値: localhost                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| -MergeDesktops                    | XMLファイルに出力されたデスクトップグループを、XenDesktop 7.xサイトの同名のデリバリーグループに統合 (つまりデスクトップを追加) します。割。このパラメーターを指定しない場合、XenDesktop 7.xサイトの既存のデリバリーグループにはデスクトップが追加されません。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| -SkipMachinePolicy                | サイトレベルの設定を保持するマシンポリシーの作成を省略します。このパラメーターを指定せず、サイトのマシンポリシーが存在する場合、スクリプトの                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| -WhatIf                           | インポート処理によりXenDesktop 7.xサイトに加えられる変更や追加を事前に確認することができます。このパラメーターを指定すると、ログファイルに多                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| -LogFilePath <path>               | ログファイルのフルパスを指定します。ログファイルには、インポート処理によりXenDesktop 7.xサイトに加えられたすべての変更内容が出力されます。ラ                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| -?または-help                        | 各パラメーターについての説明を表示します。データはインポートされません。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

XMLファイルのポリシーデータはすべてインポートされます。ただし、何らかの問題がある場合にはいずれのポリシーもインポートされません。多くの設定を含んでいる多数のポリシーのインポートには数時間かかることがあります。

スクリプトの実行が完了すると、「Done」というメッセージが表示されます。XMLファイルからのデータのインポートに成功したら、追加のエクスポートやインポートを行ったり、すべてのデーをインポートした後の「移行後のタスク」を実行したりできます。

# 移行後のタスク

Jan 04, 2017

XenDesktop 4ファームからXenDesktop 7.xサイトへのデータのインポートが完了したら、新しいXenDesktop 7.xサイトを実稼働環境で使用する前に以下のタスクを実行する必要があります。

- Virtual Delivery Agent (VDA) をアップグレードします。Controller、Studio、またはDirectorをアップグレードする前にVDAをアップグレードしておくことをお勧めします。
  - Windows VistaおよびWindows XPの場合、XenDesktop 5.6Feature Pack 1のVirtual Desktop Agentにアップグレードします。
  - Windows 7の場合、XenDesktop 7.xのVirtual Delivery Agentにアップグレードします。
- 必要に応じて、XenDesktop 7.xサイトに必要な管理者を作成します。
- ユーザーデバイスをアップデートします。ユーザーデバイス上のCitrix Receiverを最新のバージョンにアップデートすることをお勧めします。これにより、Hotfixによる修正が適用され、新しい機能がサポートされます。
- インポートしたデスクトップでレジストリベースのController検出が使用されるように変更して、次のいずれかの方法でXenDesktop 7.xのControllerを指定します。
  - レジストリを手動で編集して不要な組織単位 (OU) のGUIDレジストリエントリを削除し、ListOfDDCsレジストリエントリを追加します。
  - Active Directoryポリシー (GPMC.msc) で、Controllerの一覧をデスクトップに配信するためのマシンポリシーを設定します。Studioでこの設定を構成することはできません。

XenDesktop 7.xのデフォルトではレジストリベースのController検出が使用されますが、Active Directoryベースの検出も使用できます。

- オプションとして、XenDesktopのレジストリベースの登録に関するベストプラクティス ([CTX133384](#)) を参照して、以下のレジストリキーを設定します。
  - HeartbeatPeriodMS
  - PrepareSessionConnectionTimeoutSec
  - MaxWorkers
  - DisableActiveSessionReconnect
  - ControllersGroupGuid

XenDesktop 7.xへの移行後にこれらのレジストリキーを設定しない場合、XenDesktop 7.xのデフォルトの設定が適用されます。

- XenDesktop 4でのXMLファイルの生成時にメンテナンスモードのマシンがあった場合は、メンテナンスモードを解除します。
- XenDesktop 7.xの設定をチェックします。特に、XenDesktop 4でPortICAConfig XMLファイルを変更していた場合は、移行後の設定内容について確認します。
- 移行したすべてのコンポーネントについて、移行が正しく行われたことを確認します。

# 移行例

Jan 04, 2017

## 例1：単一の大規模XenDesktop 4ファームをXenDesktop 7サイトに移行する

ここでは、使用中の単一XenDesktop 4ファームで説明します。このXenDesktop 4ファームには50のデスクトップグループが構成されており、各グループに平均100のデスクトップが追加されています。XenDesktop 4デスクトップはProvisioning Services (PVS) でプロビジョニングされ、仮想マシンはVMware ESXハイパーバイザー上で動作しています。すべての仮想マシンにXenDesktop 4のVDAがインストールされています。

### 移行手順

1. すべてのXenDesktop 4 VDAをXenDesktop 5.6 Feature Pack 1バージョンのVDAにアップグレードします。これにより、VDAをXenDesktop 4のControllerとXenDesktop 7のDelivery Controllerの両方に登録できます。
  - Windows 7のVDAについては、「[仮想マシンまたはブレードコンピューター上のVirtual Desktop Agentのアップグレード](#)」を参照してください。
  - Windows XPおよびWindows Vista VDAについては、「[Windows XPまたはWindows Vista上のVirtual Desktop Agent](#)」を参照してください。
2. XenDesktop 4ファームにログオンしているユーザーがないことを確認します。
3. すべてのマシンがメンテナンスモードであることを確認します。
4. XenDesktop 4ファームでエクスポートツール (XdExport) を実行します。
5. XenDesktop 7の各コンポーネントをインストールします。
  1. Studioを使用して完全な実稼働環境サイトを作成します。
  2. Provisioning Servicesを使用している場合は、Provisioning Servicesサーバーおよびエージェントをアップグレードします。
  3. ライセンスサーバーおよび関連ライセンスをアップグレードします。
6. インポートツール (XdImport) の圧縮を解除して、XenDesktop 7 Controllerのローカルディレクトリ上に抽出します。
7. 手順4.で生成したエクスポートツールのXMLファイル (XdSettings.xml) を上記のローカルディレクトリにコピーします。
8. StudioでXenDesktop 7サイトのルートノードのPowerShellコンソールを開き、PowerShellセッションを起動します。
9. ハイパーバイザーの資格情報およびコピーしたXMLファイルのパスを指定してインポートツール (XdImport) を実行します。
10. Studioの [管理者] ノードで管理者設定を再作成します。詳しくは、「[管理権限の委任](#)」を参照してください。
11. インポートされたデスクトップでレジストリベースのController検出が使用されるようにして、新しいXenDesktop 7のControllerを指定します。
12. Windows 7上のVDAについては、それらのVDAをXenDesktop 7 VDA for Windows Desktop OSにアップグレードすることをお勧めします。これにより、XenDesktop 7のすべての機能を使用できるようになります。  
カタログまたはデリバリーグループのマシンのVDAソフトウェアをXenDesktop 7にアップグレードしたら、「[マシンカタログの管理](#)」および「[デリバリーグループでの設定の管理](#)」の説明に従ってカタログとデリバリーグループをアップグレードします。
13. デリバリーグループのメンテナンスモードを解除します。
14. Web Interfaceで提供していたデスクトップがStoreFrontで提供されるように構成します。詳しくは、StoreFrontのドキュメントを参照してください。

## 例2：XenDesktop 4ファームの一部をXenDesktop 7.1サイトに移行する

この例では、デスクトップをいくつかのグループに分けて段階的に移行します。ここでは、使用中の単一XenDesktop 4ファームと、作成済みで使用中の単一XenDesktop 7.1サイトで説明します。このXenDesktop 4ファームには50のデスクトッ

ブグループが構成されており、各グループに平均100のデスクトップが追加されています。XenDesktop 4デスクトップはProvisioning Servicesでプロビジョニングされ、仮想マシンはCitrix XenServerハイパーバイザー上で動作しています。すべての仮想マシンにXenDesktop 4のVDAがインストールされています。

### 移行手順

1. XenDesktop 4ファームでエクスポートツールを実行します。
  1. XenDesktop 4ファームの任意のController上でエクスポートツール (XdExport) の圧縮を解除します。
  2. Citrix管理者アカウントで、パラメーターを指定せずにエクスポートツールを実行します。
2. 生成されたXMLファイルを開き、移行対象のグループおよびデスクトップ以外のエントリを削除します。
3. XenDesktop 4ファームで、移行対象のデスクトップにログオンしているユーザーがいないことと、それらのデスクトップがメンテナンスモードであることを確認します。
4. インポートツール (XdImport) の圧縮を解除して、XenDesktop 7.1 Delivery Controllerのローカルディレクトリ上に抽出します。
5. 編集したXMLを上記のローカルディレクトリにコピーします。
6. StudioでXenDesktop 7.1サイトのルートノードのPowerShellコンソールを開き、PowerShellセッションを起動します。
7. ハイパーバイザーの資格情報およびコピーしたXMLファイルのパスを指定してインポートツール (XdImport) を実行します。
8. Studioの [管理者] ノードで管理者設定を再作成します。詳しくは、「[管理権限の委任](#)」を参照してください。
9. インポートされたデスクトップでレジストリベースのController検出が使用されるようにして、新しいXenDesktop 7.1のControllerを指定します。
10. すべてのVDAを適切なバージョンにアップグレードします。
  - Windows 7 VDAの場合は、以下を行います。
    - 「[仮想マシンまたはブレードコンピューター上のVirtual Desktop Agentのアップグレード](#)」を参照して、XenDesktop 7バージョンのVirtual Delivery Agentにアップグレードします。
    - カタログまたはデリバリーグループのマシンのすべてのVDAソフトウェアをXenDesktop 7にアップグレードしたら、「[マシンカタログの管理](#)」および「[デリバリーグループでの設定の管理](#)」の説明に従ってカタログとデリバリーグループをアップグレードします。
    - Windows XPおよびWindows Vista VDAについては、XenDesktop 5.6 Feature Pack 1バージョンのVDAにアップグレードします。「[Windows XPまたはWindows Vista上のVirtual Desktop Agent](#)」を参照してください。
11. デリバリーグループのメンテナンスモードを解除します。
12. Web Interfaceで提供していたデスクトップがStoreFrontで提供されるように構成します。詳しくは、StoreFrontのドキュメントを参照してください。

# 移行される項目と移行されない項目

Jan 04, 2017

## 移行される項目

次の表は、このリリースへの移行時に各種データがどのように移行されるかについて説明しています。特に明記されていない限り、以下の種類のデータがインポートされます。

| データの種類       | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| デスクトップグループ   | <p>「デスクトップグループ」は、このリリースでは「デリバリーグループ」と呼ばれます。デスクトップグループのアイコンはエクスポートされません。</p> <p>XenDesktop 4のDefaultEncryptionLevelがBasicでない場合、SecureIcaRequiredはTrueに設定されます。</p> <p>XenDesktop 4ファームのデスクトップグループと同じ名前のデリバリーグループが移行先のXenDesktop 7.xサイトにある場合、XenDesktop 4ファームのデスクトップグループのデスクトップをXenDesktop 7.xサイトのデリバリーグループに追加できます。</p> <p>これを行うには、インポートツールの実行時にMergeDesktopsパラメーターを指定します。XenDesktop 7.xのデリバリーグループの設定が、XenDesktop 4のデスクトップグループの設定で上書きされることはありません。このパラメーターを指定せず、XMLファイルで定義されているものと同じ名前のグループがある場合は、エラーメッセージが表示され、データはインポートされません。</p> |
| デスクトップ       | <p>プライベートデスクトップをランダムデリバリーグループに追加することはできません。ランダムデスクトップを静的デリバリーグループに追加することはできません。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| マシン          | <p>マシンは4つのマシンカタログにインポートされます。インポートツールにより、XenDesktop 7.xサイトに以下のマシンカタログが作成されます。</p> <ul style="list-style-type: none"><li>● インポートされた既存のランダム（プールされた仮想マシン用）</li><li>● インポートされた既存の静的（割り当てられた仮想マシン用）</li><li>● インポートされた物理的ランダム（プールされたPCまたはブレード用）</li><li>● インポートされた物理的静的（プライベートPCまたはブレード用）</li></ul> <p>これ以降に実行されるいずれのマシンのインポートにも、同じ4つのマシンカタログが使用されます。</p>                                                                                                                                                                             |
| プール管理<br>プール | <p>複数プールのプール、およびスケジュールを含むアイドルプール設定が含まれます。</p> <ul style="list-style-type: none"><li>● PeakBufferSizePercentは、デフォルトで10%に設定されます。</li><li>● OffPeakBufferSizePercentは、デフォルトで10%に設定されます。</li><li>● XenDesktop 4のBusiness days設定で未選択の日は、週末電源時間スキームの一部としてインポートされません。</li><li>● HostingXD4操作時間は、直近の分に切り上げられます。</li><li>● Start時間は、直近の時間に切り下げられます。</li><li>● End時間は、直近の時間に切り上げられます。</li></ul>                                                                                                                                           |
| ファーム         | <p>以下のファーム設定は、マシンポリシーとしてインポートされます。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 設定の種類           | <p>● LgaKeepAlive</p> <p>● AutoClientReconnect</p> <p>● SessionReliability</p>                                                                                                                                                                                                                                                                                                                                                    |
|                 | Flash Playerを有効にする設定はインポートされません。                                                                                                                                                                                                                                                                                                                                                                                                  |
| ポリシー            | <p>一部のポリシーデータがインポートされます。フィルター、設定項目、およびプリンターは、ユーザーポリシーとしてインポートされます。ユーザーポリシーのエクスポートおよびインポートについては、このドキュメントのほかの表を参照してください。</p> <ul style="list-style-type: none"> <li>● XenDesktop 4のデスクトップグループの設定から、新しいアクセスポリシー規則が作成されます。</li> <li>● インポートされるポリシーの相対的な優先度は保持されます。ただし、XenDesktop 7.xサイトの既存のポリシーよりも高い優先度が設定されます。</li> <li>● 移行元のポリシーを移行先のポリシーに統合することはできません。</li> </ul> <p>ポリシーをActive Directoryにインポートすることはできません。ポリシーは常にサイトに格納されます。</p> |
| ユーザー割り当て        |                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| ハイパーバイザー設定      | <p>XdlImportツールの必須パラメーターです。</p> <p>ハイパーバイザーアドレスはエクスポートされますが、ハイパーバイザーにアクセスするための資格情報はエクスポートされません。XenDesktop 7.xサイトにハイパーバイザー接続を作成するには、XMLファイルからアドレスを抽出し、対応する資格情報インスタンスにマップするPowerShellハッシュテーブルを作成します。次に、インポートツールのHypervisorConnectionCredentialsパラメーターでこのハッシュテーブルを指定します。詳しくは、「<a href="#">XenDesktop 4データのインポート</a>」を参照してください。</p> <p>既存のデスクトップグループやハイパーバイザー接続のハイパーバイザー設定をマージしたり更新したりすることはできません。</p>                                 |
| 管理者             | インポートされません。委任管理者に関するものも含め、管理者データはインポートされません。XenDesktop 7.xサイトで新しい管理者を作成してください。                                                                                                                                                                                                                                                                                                                                                    |
| ライセンス構成         | インポートされません。ライセンスサーバー名やエディションなどの情報も含め、ライセンス構成はインポートされません。ライセンスファイルはエクスポートされません。                                                                                                                                                                                                                                                                                                                                                    |
| デスクトップグループフォルダー | インポートされません。このリリースは、デスクトップグループフォルダーをサポートしません。デスクトップグループ名が重複する場合（XenDesktop 4ファームの異なるフォルダーに同名のグループがある場合など）、XMLファイルを編集してからインポートしないと処理が停止します。                                                                                                                                                                                                                                                                                         |
| レジス             | インポートされません。レジストリキーの実装については、 <a href="#">移行後のタスク</a> を参照してください。                                                                                                                                                                                                                                                                                                                                                                    |

| ポリシー<br>の種類 | 説明 |
|-------------|----|
|-------------|----|

## ユーザーポリシーデータ

次の表は、ユーザーポリシーデータがどのようにエクスポートおよびインポートされるかについて説明しています。

| XenDesktop 4でのカテゴリと設定                    | XMLファイル                    | XenDesktop 7.xでのカテゴリと設定                                                                                |
|------------------------------------------|----------------------------|--------------------------------------------------------------------------------------------------------|
| 帯域幅\視覚効果\セッションの制限<br>OEM仮想チャネル           | ClientOEMVCBandwidth       | インポートされません                                                                                             |
| クライアント\デバイスマッピング\そのほか<br>OEM仮想チャネルを無効にする | DisableOEMVirtualChannels  | インポートされません                                                                                             |
| ユーザーのワークスペース\タイムゾーン<br>サーバーのタイムゾーンを使用する  | DoNotUseClientLocalTime    | インポートされません                                                                                             |
| セキュリティ\暗号化レベル<br>SecureICA暗号化            | ClientSecurityRequirement  | インポートされません                                                                                             |
| 帯域幅\SpeedScreen<br>イメージアクセラレーション         | LossyCompression settings  | ICA\視覚表示\静止画<br>非可逆圧縮のレベル<br>非可逆圧縮のしきい値<br>ヘビーウェイト圧縮<br>ICA\視覚表示\動画<br>プログレッシブ圧縮のレベル<br>プログレッシブ圧縮のしきい値 |
| 帯域幅\視覚効果<br>デスクトップの壁紙を無効にする              | TurnOffWallpaper           | ICA\デスクトップUI<br>デスクトップの壁紙                                                                              |
| 帯域幅\視覚効果                                 | TurnOffMenuWindowAnimation | ICA\デスクトップUI                                                                                           |



| XenDesktop 4でのカテゴリと設定                   | XMLファイル                                    | XenDesktop 7.xでのカテゴリと設定                   |
|-----------------------------------------|--------------------------------------------|-------------------------------------------|
| 帯域幅\視覚効果<br>ドラッグ中にウィンドウの内容を表示する機能を無効にする | DoNotShowWindowContentsWhileDragging       | ICA\デスクトップUI<br>ドラッグ中にウィンドウの内容を表示する       |
| 帯域幅\視覚効果\セッションの制限<br>オーディオ              | ClientAudioBandwidth__AllowedBandWidth     | ICA\帯域幅<br>オーディオリダイレクトの最大帯域幅 (Kbps)       |
| 帯域幅\視覚効果\セッションの制限<br>クリップボード            | ClientClipboardBandwidth__AllowedBandWidth | ICA\帯域幅<br>クリップボードリダイレクトの最大帯域幅 (Kbps)     |
| 帯域幅\視覚効果\セッションの制限<br>COMポート             | ClientComBandwidth__AllowedBandWidth       | COMポートのリダイレクトは、XenDesktop 7.xでは使用しないでください |
| 帯域幅\視覚効果\セッションの制限<br>ドライブ               | ClientDriveBandwidth__AllowedBandWidth     | ICA\帯域幅<br>ファイルリダイレクトの最大帯域幅 (Kbps)        |
| 帯域幅\視覚効果\セッションの制限<br>LPTポート             | ClientLptBandwidth__AllowedBandWidth       | LPTポートのリダイレクトは、XenDesktop 7.xでは使用しないでください |
| 帯域幅\視覚効果\セッションの制限<br>セッション全体            | OverallBandwidth__AllowedBandWidth         | ICA\帯域幅<br>セッション全体の最大帯域幅                  |
| 帯域幅\視覚効果\セッションの制限<br>プリンター              | LimitPrinterBandWidth__AllowedBandWidth    | ICA\帯域幅<br>プリンターリダイレクトの最大帯域幅 (Kbps)       |
| クライアント\デバイスマッピング<br>\オーディオ<br>マイク       | ClientAudioMicrophone__TurnOn              | ICA\オーディオ<br>クライアントマイクリダイレクト              |

|                                                            |                                               |                                                   |
|------------------------------------------------------------|-----------------------------------------------|---------------------------------------------------|
| <b>XenDesktop 4でのカテゴリと設定</b><br>クライアント\デバيسマッピング<br>\オーディオ | <b>XMLファイル</b><br>ClientAudioQuality__Quality | <b>XenDesktop 7.xでのカテゴリと設定</b><br>ICA\オーディオ<br>音質 |
| 音質                                                         |                                               |                                                   |
| クライアント\デバيسマッピング<br>\オーディオ<br><br>スピーカーを無効にする              | DisableClientAudioMapping                     | ICA\オーディオ<br><br>クライアントオーディオリ<br>ダイレクト            |
| クライアント\デバيسマッピング\ド<br>ライブ<br><br>接続                        | ConnectClientDriveAtLogon__TurnOn             | ICA\ファイルリダイレクト<br><br>ドライブへの自動接続                  |
| クライアント\デバيسマッピング\ド<br>ライブ<br><br>フロッピーディスクドライブ             | DisableClientDriveMapping__DisableFloppyDrive | ICA\ファイルリダイレクト<br><br>クライアント側フロッピー<br>ドライブ        |
| クライアント\デバيسマッピング\ド<br>ライブ<br><br>ハードディスクドライブ               | DisableClientDriveMapping__DisableHardDrive   | ICA\ファイルリダイレクト<br><br>クライアント側固定ライ<br>ブ            |
| クライアント\デバيسマッピング\ド<br>ライブ<br><br>CD-ROMドライブ                | DisableClientDriveMapping__DisableCdrom       | ICA\ファイルリダイレクト<br><br>クライアント側光学式ドラ<br>イブ          |
| クライアント\デバيسマッピング\ド<br>ライブ<br><br>リモートドライブ                  | DisableClientDriveMapping__DisableRemote      | ICA\ファイルリダイレクト<br><br>クライアント側ネットワー<br>クドライブ       |
| クライアント\デバيسマッピング\ド<br>ライブ<br><br>USBディスクドライブ               | DisableClientDriveMapping__DisableUSB         | ICA\ファイルリダイレクト<br><br>クライアント側リムーバブル<br>ドライブ       |
| クライアント\デバيسマッピング\ド<br>ライブ\最適化<br><br>非同期書き込み               | CDMAsyncWrites                                | ICA\ファイルリダイレクト<br><br>非同期書き込みを使用する                |
|                                                            |                                               |                                                   |

|                                                                                            |                                                  |                                                                             |
|--------------------------------------------------------------------------------------------|--------------------------------------------------|-----------------------------------------------------------------------------|
| <p>XenDesktop 4でのカテゴリを設定<br/>のほかに</p> <p>クライアント\デバイスマッピング<br/>\ポート</p> <p>COMポートを無効にする</p> | <p>DisableClientClipboardMapping<br/>XMLファイル</p> | <p>ICA<br/>XenDesktop 7.xでのカテゴ<br/>リと設定</p> <p>クライアントクリップポー<br/>ドリダイレクト</p> |
| <p>クライアント\デバイスマッピング<br/>\ポート</p> <p>COMポートを無効にする</p>                                       | <p>DisableClientCOMPortMapping</p>               | <p>COMポートのリダイレクト<br/>は、XenDesktop 7.xでは使<br/>用しないでください</p>                  |
| <p>クライアント\デバイスマッピング<br/>\ポート</p> <p>LPTポートを無効にする</p>                                       | <p>DisableClientLPTPortMapping</p>               | <p>LPTポートのリダイレクト<br/>は、XenDesktop 7.xでは使<br/>用しないでください</p>                  |
| <p>クライアント\デバイスマッピング<br/>\USB</p> <p>USB</p>                                                | <p>RemoteUSBDevices__DisableRemoteUSBDevices</p> | <p>ICA\USBデバイス</p> <p>クライアントUSBデバイス<br/>リダイレクト</p>                          |
| <p>印刷\クライアントプリンター</p> <p>自動作成</p>                                                          | <p>ConnectClientPrinterAtLogon__Flag</p>         | <p>ICA\印刷\クライアントプリ<br/>ンター</p> <p>クライアントプリンターを<br/>自動作成する</p>               |
| <p>印刷\クライアントプリンター</p> <p>従来のクライアントプリンター</p>                                                | <p>LegacyClientPrinters__TurnOn</p>              | <p>ICA\印刷\クライアントプリ<br/>ンター</p> <p>クライアントプリンター名</p>                          |
| <p>印刷\クライアントプリンター</p> <p>プリンタープロパティの保存</p>                                                 | <p>ModifiedPrinterProperties__WriteMethod</p>    | <p>ICA\印刷\クライアントプリ<br/>ンター</p> <p>プリンタープロパティの保<br/>存</p>                    |
| <p>印刷\クライアントプリンター</p> <p>印刷ジョブの送信</p>                                                      | <p>ClientPrintingForNetworkPrinter__TurnOn</p>   | <p>ICA\印刷\クライアントプリ<br/>ンター</p> <p>プリントサーバーへの直接<br/>接続</p>                   |
| <p>印刷\クライアントプリンター</p> <p>クライアントプリンターのマッピン</p>                                              | <p>DisableClientPrinterMapping</p>               | <p>ICA\印刷</p> <p>クライアントプリンターリ</p>                                           |

| 機能を無効にする<br>XenDesktop 4でのカテゴリと設定         | XMLファイル                                                                              | ダイレクト<br>XenDesktop 7.xでのカテゴリと設定           |
|-------------------------------------------|--------------------------------------------------------------------------------------|--------------------------------------------|
| 印刷\ドライバー<br><br>ネイティブプリンタードライバーの自動インストール  | PrintDriverAutoInstall__TurnOn                                                       | ICA\印刷\ドライバー<br><br>付属のプリンタードライバーの自動インストール |
| 印刷\ドライバー<br><br>ユニバーサルドライバー               | ClientPrintDriverToUse                                                               | ICA\印刷\ドライバー<br><br>ユニバーサル印刷の使用            |
| 印刷\セッションプリンター<br><br>セッションプリンター           | NetworkPrinters                                                                      | ICA\印刷<br><br>セッションプリンター                   |
| 印刷\セッションプリンター<br><br>クライアントのデフォルトプリンターの選択 | DefaultToMainClientPrinter__NetworkDefault<br><br>DefaultToMainClientPrinter__TurnOn | ICA\印刷<br><br>デフォルトプリンター                   |

#### 移行されない項目

XenDesktop 4の一部のコンポーネントは、このリリースでサポートされません。以下の項目は、移行されません。

- **Virtual Delivery Agent** — XenDesktop 4の仮想デスクトップをXenDesktop 7.x Delivery Controllerで管理できるようにするには、VDAをXenDesktop 5.x以降のVDAにアップグレードする必要があります。VDAのアップグレードについては、「[移行後のタスク](#)」を参照してください。
- **Controller** — 新しいControllerサーバーを展開する必要があります。XenDesktop 4のControllerをXenDesktop 7.xサイトにアップグレードすることはできません。XenDesktop 7.xサイトをXenDesktop 4ファームに追加したりXenDesktop 4 ControllerをXenDesktop 7.xサイトに追加したりすることはできません。また、バージョンによってサーバー要件が異なります。XenDesktop 4にはWindows Server 2003が必要であり、XenDesktop 7.xにはより新しいバージョンのWindows Serverが必要です。
- **Web Interface** — XenDesktop 7.xではStoreFrontを使用することをお勧めします。インストールとセットアップについては、StoreFrontのドキュメントを参照してください。XenDesktopのインストール時にWeb Interfaceが検出されるとStoreFrontがインストールされますが、Web Interfaceは削除されません。
- **Active Directory組織単位 (OU) 構成** — 1つの組織単位 (OU) をファーム間またはサイト間、またはファームとサイト間で共有することはサポートされていません。新しいサイトを作成して、デフォルトのレジストリベースのController検出の代わりにActive DirectoryベースのController検出を使用する場合は、そのための新しいOUを作成する必要があります。
- **PortICAConfig XMLファイル** — このファイルに対するデフォルトの設定を変更している場合は、新しいサイトでグループポリシーオブジェクトを使用してこれらの設定を構成する必要がある場合があります。
- **XenDesktop 4 Service Pack 1**で提供される構成ログ設定。
- **Provisioning Services**関連のデータ。
- アプリケーション。
- **Controller**の一覧。
- **NetScaler Gateway**。
- イベントログの調整設定。

# セキュリティ

Jan 04, 2017

 Citrix XenAppおよびXenDesktopのセキュリティの概要

XenAppおよびXenDesktopでは、セキュリティニーズに合わせて環境をカスタマイズできる、セキュアバイデザイン（セキュリティに配慮した設計）ソリューションが提供されます。

モバイルワーカーへの対応でIT部門が直面するセキュリティ上の問題に、データの紛失や盗難があります。XenAppおよびXenDesktopでは、アプリケーションとデスクトップがホストされ、すべてのデータがデータセンターに保持されるため、機密データや知的財産がエンドポイントデバイスから安全に分離されます。データ転送を許可するポリシーを有効にしている場合でも、すべてのデータが暗号化されます。

また、XenDesktopおよびXenAppのデータセンターでは、一元的な監視と管理サービスを利用できるため、インシデント対応が容易になります。Directorでは、ネットワーク経由でアクセスされたデータを監視して分析できます。また、Studioではデータセンターにパッチを適用して多くの脆弱性を解決できるため、エンドユーザーデバイスごとにローカルで問題を解決する必要がありません。

XenAppおよびXenDesktopでは、一元化された監査記録を使用して、どのアプリケーションやデータにどのユーザーがアクセスしたかを判別できるため、監査と法規制順守も簡素化されます。Directorでは、構成ログとOData APIにアクセスして、システムに適用された更新とユーザーのデータ使用状況に関する履歴データが収集されます。

管理権限の委任によって、管理者の役割を設定して、XenDesktopおよびXenAppへのアクセスを詳細に制御できます。これにより、ほかの管理者のアクセス権は制限したままで、特定の管理者に対してタスク、操作、およびスコープへの完全なアクセス権を組織内で柔軟に付与できます。

XenAppおよびXenDesktopでは、ローカルレベルから組織単位レベルまで、ネットワークのさまざまなレベルでポリシーを適用してユーザーを制御できます。このポリシー制御によって、ユーザー、デバイス、またはユーザーやデバイスのグループが実行できる操作（接続、印刷、コピーと貼り付け、ローカルドライブのマップ）を指定できるため、社外作業員に対するセキュリティ上の問題を最小限に抑えることができます。Desktop Lock機能を使用すると、エンドユーザーデバイスのローカルのオペレーティングシステムにアクセスできないようにして、エンドユーザーによる使用を仮想デスクトップのみに制限することも可能です。

管理者は、Controllerで、またはエンドユーザーとVDA（Virtual Delivery Agent）間でSSL（Secure Sockets Layer）セキュリティプロトコルが使用されるように構成して、XenAppまたはXenDesktopのセキュリティを強化できます。また、TLS（Transport Layer Security）セキュリティプロトコルを有効にして、TCP/IP接続に対してサーバー認証、データストリームの暗号化、およびメッセージの整合性チェックが行われるようにすることもできます。

さらに、XenAppおよびXenDesktopでは、Windowsや特定のアプリケーションでの複数要素認証がサポートされています。複数要素認証を使用して、XenAppおよびXenDesktopで配信されるすべてのリソースを管理することもできます。以下の認証方法を使用できます。

- トークン
- スマートカード
- RADIUS
- Kerberos
- 生体認証

XenDesktopは、ID管理からウイルス対策ソフトウェアまで、さまざまなサードパーティセキュリティソリューションを統合

できます。サポートされている製品の一覧については、<http://www.citrix.com/ready>を参照してください。

XenAppおよびXenDesktopの一部リリースは、情報セキュリティ国際評価基準（コモンクライテリア）の認定を受けています。これらの基準の一覧については、<http://www.commoncriteriaportal.org/cc/>を参照してください。

#### 関連トピック

- [セキュリティに関するベストプラクティスと考慮事項](#)
- [管理権限の委任](#)
- [スマートカード](#)
- [SSL](#)
- [Desktop Lock](#)

# セキュリティに関するベストプラクティスと考慮事項

Jan 04, 2017

このドキュメントでは次のことを説明しています。

- このリリースを使用するときの一般的なセキュリティに関する推奨事項、およびこのリリースと従来のコンピューター環境でのセキュリティに関する相違点
- ユーザーアカウントの管理
- ユーザー権限の管理
- ログオン権限の管理
- ユーザー権利の構成
- サービス設定の構成
- 展開シナリオとそのセキュリティ設定
- リモートPCアクセスのセキュリティに関する考慮事項

組織によっては、法的規制の要件を満たすために特定のセキュリティ基準への準拠が要求される場合があります。このようなセキュリティ基準は変更されることがあるため、ここでは説明しません。セキュリティ基準とCitrix製品に関する最新情報については、<http://www.citrix.com/security/>を参照してください。

## セキュリティに関する推奨事項

セキュリティパッチを適用して、環境内にあるすべてのマシンを最新の状態にします。この製品の利点の1つは、シンクライアントをターミナルとして使用することによってこの作業を簡略化できることです。

環境内にあるすべてのマシンを、アンチウイルスソフトウェアで保護します。

環境内にあるすべてのマシンを、境界ファイアウォール（必要に応じてエンクレープ境界を含む）で保護します。

従来の環境を新しいバージョンに移行する場合は、既存の境界ファイアウォールを移動するか、新しい境界ファイアウォールを追加する必要があります。たとえば、従来のクライアントとデータセンター内のデータベースサーバーとの間に境界ファイアウォールがあるとします。このリリースを使用するときは、仮想デスクトップおよびユーザーデバイスと、データセンター内のデータベースサーバーおよびDelivery Controllerとの間に境界ファイアウォールを設定する必要があります。したがって、データベースサーバーとControllerを含むエンクレープをデータセンター内に作成することを検討します。また、ユーザーデバイスと仮想デスクトップ間のセキュリティについても考慮する必要があります。

環境内にあるすべてのマシンは、パーソナルファイアウォールで保護する必要があります。コアコンポーネントとVirtual Delivery Agent (VDA) をインストールするときにWindows Firewallサービスが検出された場合は（ファイアウォールが無効であったとしても）、コンポーネントと機能の通信に必要なポートが自動的に開放されるように設定できます。また、そのファイアウォールポートを手作業で構成することもできます。Windows以外のファイアウォールを使用している場合は、手作業でファイアウォールを構成する必要があります。

注：TCPポート1494および2598はICAおよびCGPに使用され、ファイアウォールで開放されているため、データセンター外のユーザーはこれらのポートにアクセスできます。管理インターフェイスが不注意で開いたままになって攻撃を受ける可能性を避けるため、これらのTCPポートをほかの目的で使用しないでください。ポート1494および2598は、Internet Assigned Number Authority (<http://www.iana.org/>を参照) に正規登録されています。

すべてのネットワーク通信が正しく保護され、セキュリティポリシーに従って暗号化されている必要があります。IPSecを使用して、Microsoft Windowsコンピューターの間でのすべての通信を保護できます。その方法について詳しくは、使用するオペレーティングシステムのドキュメントを参照してください。さらに、ユーザーデバイスとデスクトップ間の通信は、デフォルトで128ビット暗号化を行うCitrix SecureICAで保護できます。SecureICAは、デリバリーグループの作成または更新時に構成できます。詳しくは、「[基本設定の変更](#)」を参照してください。

# ユーザーアカウントの管理

Virtual Delivery Agent (VDA) インストール時に、App-V公開コンポーネントをインストールするオプションが選択された場合、あるいはこの機能を後で追加する場合は、ローカル管理アカウントCtxAppVCOMAdminがVDAに追加されます。App-V公開機能を使用する場合は、このアカウントに変更を加えないようにします。App-V公開機能を使用する必要がない場合は、インストール時にこのアカウントを選択しないでください。後でApp-V公開機能が不要になった場合には、機能を無効にしたり削除したりすることができます。

## ユーザー権限の管理

ユーザーには、必要な権限だけを付与します。デスクトップのユーザーには、Microsoft Windowsでの権限（グループポリシーの「ユーザー権利の割り当て」およびグループメンバーシップ）がそのまま適用されます。このリリースの利点の1つは、仮想デスクトップが格納されているコンピューターに対する物理的な制御を許可せずに、デスクトップに対するユーザーの管理権限を付与できることです。

デスクトップに対する管理権限を計画するときは、以下の点に注意してください。

- デフォルトでは、権限を持たないユーザーがデスクトップに接続すると、ユーザーデバイスのタイムゾーンではなく、そのデスクトップを実行しているシステムのタイムゾーンが表示されます。デスクトップの使用時にローカルの時刻が表示されるようにする方法については、「[基本設定の変更](#)」を参照してください。
- デスクトップの管理者権限を持つユーザーは、そのデスクトップを完全に制御できます。デスクトップが専用デスクトップではなくプールデスクトップの場合、管理者権限を持つユーザーはそのデスクトップのすべてのユーザー（将来のユーザーを含む）に信頼されている必要があります。このため、プールデスクトップのすべてのユーザーは、この状況によってデータのセキュリティに永続的な危険性が存在することを認識する必要があります。これは、単一ユーザーに対してのみ割り当てられるデスクトップには当てはまりません。つまり、このユーザーはほかのデスクトップの管理者になることができません。
- 通常、デスクトップの管理者であるユーザーはそのデスクトップにソフトウェアをインストールできます。インストールできるソフトウェアには悪意のあるものも含まれます。またユーザーが、そのデスクトップに接続しているすべてのネットワーク上のトラフィックを監視または制御することも可能です。

## ログオン権限の管理

ユーザーアカウントとコンピューターアカウントの両方にログオン権限が必要です。Microsoft Windowsの権限では、ログオン権限は引き続き、「ユーザー権限の割り当て」で権限を設定し「グループポリシー」でグループメンバーシップを設定するという通常の方法で、デスクトップに適用されます。

Windowsのログオン権限には次の種類があります。ローカルログオン、リモートデスクトップサービスを使ったログオン、ネットワーク経由でのログオン（ネットワーク経由でコンピューターへアクセス）、バッチジョブとしてログオン、サービスとしてログオン。

コンピューターアカウントでは、必要なログオン権限だけをコンピューターに付与します。次のアカウントに、ログオン権限「ネットワーク経由でコンピューターへアクセス」が必要です。

- VDAで、Delivery Controllerのコンピューターアカウント
- Delivery Controllerで、VDAのコンピューターアカウント。「[Active Directory OUベースのController検出](#)」を参照してください。
- StoreFrontサーバーで、同じStoreFrontサーバーグループ内の他のサーバーのコンピューターアカウント



ユーザーアカウントでは、必要なログオン権限だけをユーザーに付与します。

Microsoftによると、デフォルトでRemote Desktop Usersグループに [リモート デスクトップ サービスを使ったログオンを許可] でログオン権限が付与されています (ドメインコントローラを除く)。

組織のセキュリティポリシーによっては、このグループがこのログオン権限から除外されることを明示的に設定している場合もあります。次の方法を検討してください。

- Virtual Delivery Agent (VDA) for Server OSはMicrosoftリモートデスクトップサービスを使用します。Remote Desktop Usersグループを制限されたグループとして構成し、Active Directoryグループポリシー経由でグループのメンバーシップを制御できます。詳しくは、Microsoft社のドキュメントを参照してください。
- VDA for Desktop OSを含むXenAppおよびXenDesktopの他のコンポーネントでは、Remote Desktop Usersグループは必要ありません。このため、これらのコンポーネントではRemote Desktop Usersグループにログオン権限 [リモート デスクトップ サービスを使ったログオンを許可] の必要はなく、削除できます。さらに、以下を確認します。
  - リモートデスクトップサービスでこれらのコンピューターを管理する場合、すべての必要な管理者が既にAdministratorグループのメンバーであることを確認してください。
  - リモートデスクトップサービスでこれらのコンピューターを管理しない場合、コンピューター上でリモートデスクトップサービスを無効にすることを検討してください。

ユーザーとグループをログオン権限 [リモートデスクトップサービスによるログオンを拒否] に追加することは可能ですが、ログオン権限の拒否の使用は、通常推奨されません。詳しくは、Microsoft社のドキュメントを参照してください。

## ユーザー権利の構成

Delivery Controllerをインストールすると、次のWindowsサービスが作成されます。

- Citrix AD Identity Service (NT SERVICE\CitrixADIdentityService) : 仮想マシンのMicrosoft Active Directoryコンピューターアカウントを管理します。
- Citrix Analytics (NT SERVICE\CitrixAnalytics) : Citrixが使用するサイト構成の使用状況情報の収集がサイト管理者によって承認されている場合、この情報を収集します。その後、製品の改善に役立てるために、この情報をCitrixに送信します。
- Citrix App Library (NT SERVICE\CitrixAppLibrary) : AppDiskの管理とプロビジョニング、AppDNA統合、およびApp-Vの管理をサポートします。
- Citrix Broker Service (NT SERVICE\CitrixBrokerService) : ユーザーが使用できる仮想デスクトップやアプリケーションを選択します。
- Citrix Configuration Logging Service (NT SERVICE\CitrixConfigurationLogging) : すべての構成の変更と、管理者がサイトに対して行ったそのほかの状態の変更を記録します。
- Citrix Configuration Service (NT SERVICE\CitrixConfigurationService) : 共有される構成のサイト全体のリポジトリです。
- Citrix Delegated Administration Service (NT SERVICE\CitrixDelegatedAdmin) : 管理者に与えられた権限を管理します。
- Citrix Environment Test Service (NT SERVICE\CitrixEnvTest) : そのほかのDelivery Controllerサービスのセルフテストを管理します。
- Citrix Host Service (NT SERVICE\CitrixHostService) : XenAppまたはXenDesktop環境で使用されているハイパーバイザーインフラストラクチャに関する情報を保存します。また、コンソールで使用される、ハイパーバイザープールのリソースを列挙する機能を提供します。
- Citrix Machine Creation Service (NT SERVICE\CitrixMachineCreationService) : デスクトップ仮想マシンの作成をオーケストレーションします。
- Citrix Monitor Service (NT SERVICE\CitrixMonitor) : XenAppまたはXenDesktopの測定基準を収集し、履歴情報を保存して、トラブルシューティングのためのクエリインターフェイスと各種のレポートツールを提供します。
- Citrix Storefront Service (NT SERVICE\CitrixStorefront) : StoreFrontの管理をサポートします (StoreFrontコンポーネ

ント自体には含まれていません)。

- Citrix Storefront Privileged Administration Service (NT SERVICE\CitrixPrivilegedService) : StoreFrontの特権管理操作をサポートします (StoreFrontコンポーネント自体には含まれていません)。

Delivery Controllerをインストールすると、次のWindowsサービスも作成されます。これらは、そのほかのCitrixコンポーネントをインストールしたときにも作成されます。

- Citrix Diagnostic Facility COM Server (NT SERVICE\CdfSvc) : Citrixサポートが使用するための診断情報の収集をサポートします。
- Citrix Telemetry Service (NT SERVICE\CitrixTelemetryService) : Citrixが分析するための診断情報を収集することで、管理者が分析結果と推奨事項を確認してサイトの問題解決に役立てることができるようにします。

Citrix Storefront Privileged Administration Serviceを除く、これらのサービスには、[サービスとしてログオン]のログオン権限と[プロセスのメモリクォータの増加]、[セキュリティ監査の生成]、[プロセスレベルトークンの置き換え]の権限が付与されます。通常、これらのユーザー権利を変更する必要はありません。これらの権限はDelivery Controllerでは使用されないため、自動的に無効にされています。

## サービス設定の構成

Citrix Storefront Privileged Administration ServiceとCitrix Telemetry Serviceを除く、上述の「ユーザー権利の構成」セクションに示すDelivery Controller Windowsサービスは、ネットワークサービスIDでログオンするように構成されます。このサービス設定は変更しないでください。

Citrix Storefront Privileged Administration Serviceは、ローカルシステム (NT AUTHORITY\SYSTEM) にログオンするように構成されます。これは、通常はサービスで実行できないDelivery Controller StoreFront操作 (Microsoft IISサイトの作成など) に必要な構成です。このサービス設定は変更しないでください。

Citrix Telemetry Serviceは、このサービス自体のサービス固有のIDでログオンするように構成されます。

Citrix Telemetry Serviceは、無効にすることができます。このサービスと、既に無効にされているサービス以外のそのほかのDelivery Controller Windowsサービスは、無効にしないでください。

### 展開シナリオのセキュリティ

ユーザー環境は、組織に管理されずにユーザーにより完全に制御されるユーザーデバイス、または組織により管理されたユーザーデバイスで構成できます。通常、これら2つの環境に対するセキュリティ上の考慮事項は異なります。

- **管理されるユーザーデバイス** — 管理者または信頼されたほかの組織によって管理されるユーザーデバイスを指します。この場合、ユーザーデバイスを管理者が構成してユーザーに直接提供したり、全画面のみを実行するモードで単一のデスクトップを実行する端末を提供したりできます。管理されるユーザーデバイスに対しては、前述の一般的なセキュリティ構成を実装する必要があります。この製品の長所は、ユーザーデバイス上に最低限のソフトウェアしか必要としないという点です。

管理されるユーザーデバイスでは、仮想デスクトップの実行モードとして、全画面のみを実行するモードまたはウィンドウモードを設定できます。

- 全画面のみを実行するモードを使用する場合、ユーザーは通常の [Windowsへのログオン] 画面からユーザーデバイスにログオンします。すると、同じユーザー資格情報で自動的にこのリリースへのログオンが実行されます。
- 一方、ウィンドウモードを使用する場合、ユーザーは最初にユーザーデバイスにログオンし、次にこのリリースで提供されたWebサイトを介してこの製品にログオンします。

- **管理されていないユーザーデバイス** — 管理者または信頼された組織によって管理されていないユーザーデバイスを指します。たとえば、ユーザーが自分のデバイスを使用する場合、上記のセキュリティ上の推奨事項にユーザーが従わないことがあります。このリリースでは、このような管理されていないユーザーデバイスにも、デスクトップを安全に配信できます。ただし、これらのユーザーデバイスでも、キーロガーやそれに類似した入力攻撃を阻止するための基本的なウイルス対策が施されている必要があります。
- **データストレージの考慮事項** — このリリースを使用しているときに、ユーザーが自分のユーザーデバイスにデータを保存できないように構成できます。ただし、ユーザーが仮想デスクトップにデータを保存することを許可するかどうかも考慮する必要があります。ユーザーによるデスクトップ上へのデータ保存は推奨されません。データはファイルサーバー、データベースサーバー、またはデータが適切に保護されるそのほかのリポジトリに保存する必要があります。

デスクトップ環境は、プールデスクトップや専用デスクトップなど、さまざまな種類のデスクトップで構成される場合があります。

- ユーザーは、プールデスクトップなど、複数のユーザーで共有されるデスクトップ上にデータを保存するべきではありません。
- また、専用デスクトップでも、そのデスクトップをほかのユーザーが使用することになった場合に、保存されているデータを削除する必要があります。
- **混在バージョン環境** — アップグレード処理のある時点においては、バージョンが混在する環境は不可避なものです。ベストプラクティスに従い、異なるバージョンのCitrixコンポーネントが同時に存在する時間を最短化させます。

たとえばバージョン混在環境ではセキュリティポリシーが一律には適用されない可能性があります。

注：これは、ほかのソフトウェア製品では一般的な問題です。Active Directoryの以前のバージョンを使用すると、最近のバージョンのWindowsにはグループポリシーが部分的にしか適用されません。

次のシナリオでは、特定のバージョン混在Citrix環境で発生する可能性があるセキュリティ問題について説明します。XenAppおよびXenDesktop 7.6 Feature Pack 2のVirtual Delivery Agentを実行している仮想デスクトップへの接続にCitrix Receiver 1.7が使用されている場合、ポリシー「デスクトップとクライアント間におけるファイル転送の許可」はサイトでは有効ですが、XenAppおよびXenDesktop 7.1を実行しているDelivery Controllerによっては無効にできません。製品のより新しいバージョンでのみリリースされたポリシーは認識されません。このポリシーにより、ユーザーはファイルを自分の仮想デスクトップにアップロードしてダウンロードできます - セキュリティ問題。この問題を回避するには、Delivery ControllerあるいはStudioのスタンドアロンインスタンスをバージョン7.6 Feature Pack 2にアップグレードし、その後でグループポリシーを使ってポリシーを無効にします。または、すべての該当する仮想デスクトップでローカルポリシーを使用します。

## リモートPCアクセス

リモートPCアクセスでは、次のセキュリティ機能がサポートされます。

- スマートカードの使用がサポートされます。
- リモートセッションの間、社内のPCのモニターは非表示になります。
- リモートPCアクセスでは、すべてのキーボードおよびマウスの入力が入力セッションにリダイレクトされます (Ctrl+Alt+Delキー入力、およびUSB対応スマートカードや生体認証デバイスを除く)。
- SmoothRoamingは単一ユーザーに対してのみサポートされます。
- リモートセッションで接続していた社内のPCにローカルでアクセスを再開できるのはそのユーザーのみです。ローカルでのアクセスを再開するには、ローカルのキーボードでCtrl+Alt+Delキーを押して、リモートセッションと同じ資格情報を使ってログオンします。システムに適切なサードパーティ製の資格情報プロバイダー統合が構成されている場合は、スマートカードを挿入したり生体認証を使用したりしてローカルアクセスを再開することもできます。グループポリシーオブジェクト (GPO) やレジストリキーでユーザーの簡易切り替え機能を有効にして、このデフォルトの動作設定を上書きすることができます。

- リモートPCアクセスでは、デフォルトで単一VDAへの複数ユーザーの自動割り当てがサポートされます。XenDesktop 5.6 Feature Pack 1では、PowerShellスクリプトRemotePCAccess.ps1を使ってこの動作を上書きできました。このリリースでは、レジストリキーを使って複数ユーザーの自動割り当てを許可または禁止できます。この設定はサイト全体に適用されません。

注意：レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

自動割り当てを単一ユーザーのみに制限するには、以下の手順に従います。

1. サイトの各Controllerで、次のレジストリエントリを設定します。

HKEY\_LOCAL\_MACHINE\Software\Citrix\DesktopServer

値の名前：AllowMultipleRemotePCAssignments

種類：REG\_DWORD

値のデータ：0（複数ユーザーの割り当て無効）、1（デフォルト。複数ユーザーの割り当て有効）

2. 既存のユーザー割り当てを削除するには、SDKコマンドを使用します。これにより、VDAに単一ユーザーが割り当てられるようになります。
  1. 割り当てられているすべてのユーザーをVDAから削除するには、`$machine.AssociatedUserNames | %{ Remove-BrokerUser-Name $_ -Machine $machine}`を実行します。
  2. デリバリーグループからVDAを削除するには、`$machine | Remove-BrokerMachine -DesktopGroup $desktopGroup`を実行します。
3. 社内の物理PCを再起動します。

# 管理権限の委任

Jan 04, 2017

委任管理モデルにより、役割やオブジェクトベースの制御により、組織の管理業務の分担に基づいて柔軟に管理権限を委任することができます。あらゆる規模のサイトで委任管理機能を使用でき、展開環境が複雑化するにつれてより詳細な権限の分担を構成できます。管理権限の委任機能では、管理者、役割、およびスコープという3つの概念が使用されます。

- **管理者** — 管理者は、Active Directoryアカウントにより識別される、管理権限を持つ個人またはそのグループを示します。各管理者には、1つまたは複数の役割とスコープのペアが割り当てられます。
- **役割** — 役割は管理ジョブの機能を表し、それぞれ定義された権限が割り当てられています。たとえば、[デリバリーグループ管理者]の役割には、「デリバリーグループの作成」および「デリバリーグループからのデスクトップの削除」などの権限があります。管理者は、サイトに対して複数の役割を有することができ、1人の管理者がデリバリーグループ管理者とマシンカタログ管理者を兼ねることができます。役割には、組み込みの役割とカスタムの役割があります。

組み込みの役割は、次のとおりです。

| 役割                | 権限                                                                                                                                                                                             |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| すべての管理権限を実行できる管理者 | すべてのタスクと操作を実行できます。[すべての管理権限を実行できる管理者]の役割は、常に[すべて]のスコープとペアになります。                                                                                                                                |
| 読み取り専用管理者         | 全体的な情報に加えて、指定されたスコープのすべてのオブジェクトを表示できますが、変更することはできません。たとえば、「大阪」というスコープを作成して読み取り専用管理者に割り当てると、構成ログなどのグローバルオブジェクトと、大阪支社用のデリバリーグループなど、[大阪]スコープのオブジェクトを表示できます。ただし、この管理者は「ニューヨーク」スコープのオブジェクトを表示できません。 |
| ヘルプデスク管理者         | デリバリーグループを表示して、関連付けられているセッションとマシンを管理できます。デリバリーグループのマシンカタログやホスト情報を表示したり、デリバリーグループのマシンのセッションや電源を管理したりできます。                                                                                       |
| マシンカタログ管理者        | マシンカタログを作成および管理したり、マシンカタログにマシンをプロビジョニングしたりできます。仮想化インフラストラクチャ、Provisioning Services、および物理マシンを使用してマシンカタログを作成できます。この役割では、基本イメージを管理したりソフトウェアをインストールしたりできますが、アプリケーションやデスクトップをユーザーに割り当てることはできません。    |
| デリバリーグループ管理者      | アプリケーション、デスクトップ、およびマシンを配信したり、それらのセッションを管理したりできます。ポリシーや電源管理設定など、アプリケーションおよびデスクトップの構成を管理することもできます。                                                                                               |
| ホスト管理者            | ホスト接続およびその関連リソース設定を管理できます。マシン、アプリケーション、またはデスクトップをユーザーに配信することはできません。                                                                                                                            |

この製品の一部のエディションでは、必要に応じてカスタムの役割を作成して、より詳細な権限を委任することができます。カスタムの役割では、コンソールにおける操作またはタスク単位で権限を割り当てることができます。

- **スコープ** — 接続、マシンカタログ、デリバリーグループなど、その管理者が管理できるオブジェクトをグループ化したものです。スコープでは、組織の要件に基づいてオブジェクトをグループ化します（営業チームで使用されるデリバリーグループのセットなど）。オブジェクトを複数のスコープに含めることができます。つまり、1つまたは複数のスコープでオブジェクトをラベル付けすることができます。組み込みのスコープである「すべて」には、すべてのオブジェクトが含まれています。[すべての管理権限を実行できる管理者]の役割は、常にこのスコープとペアになります。

## 例

XYZ社は自社の部署（経理、営業、倉庫）およびそのデスクトップオペレーティングシステム（Windows 7またはWindows 8）に基づいてアプリケーションとデスクトップを管理することにしました。管理者は5つのスコープを作成し、各デリバリーグループに2つのスコープ（部署を表すスコープと使用するオペレーティングシステムを表すスコープ）を割り当てました。

次の管理者を作成しました。

| 管理者                   | 役割                     | スコープ                                             |
|-----------------------|------------------------|--------------------------------------------------|
| domain/fred           | すべての管理権限を実行できる管理者      | すべて（[すべての管理権限を実行できる管理者]の役割は、常に[すべて]スコープとペアになります） |
| domain/rob            | 読み取り専用管理者              | すべて                                              |
| domain/heidi          | 読み取り専用管理者<br>ヘルプデスク管理者 | すべて<br>営業                                        |
| domain/warehouseadmin | ヘルプデスク管理者              | 倉庫                                               |

| 管理者          | 役割                         | スコープ |
|--------------|----------------------------|------|
| domain/peter | デリバリーグループ管理者<br>マシンカタログ管理者 | Win7 |

- Fredは「すべての管理権限を実行できる管理者」で、システム内のすべてのオブジェクトを表示、編集、および削除できます。
- Robはサイト内のすべてのオブジェクトを表示できますが、それらを編集または削除することはできません。
- Heidiはすべてのオブジェクトを表示でき、[営業] スコープのデリバリーグループでヘルプデスクタスクを実行できます。これにより、[営業] スコープのデリバリーグループに割り当てられているセッションとマシンを管理できます。ただし、これらのデリバリーグループに（マシンの追加や削除などの）変更を加えることはできません。
- Active Directoryセキュリティグループwarehouseadminのすべてのメンバーは、[倉庫] スコープのマシンに対するヘルプデスクタスクを表示および実行できます。
- PeterはWindows 7の専門家ですべてのWindows 7マシンカタログを管理でき、所属している部署のスコープに関係なくWindows 7アプリケーション、デスクトップ、およびマシンを配信できます。当初、管理者はPeterを [Win7] スコープの「すべての管理権限を実行できる管理者」にしようとしたが、考え直しました。これは、「すべての管理権限を実行できる管理者」には、そのスコープに含まれていないオブジェクト（「サイト」や「管理者」など）に対する全権限が付与されるためです。

## 委任管理の使用法

一般的に、管理者数およびその権限の細分性は展開のサイズおよびその複雑度に応じて異なります。

- 小規模または検証用の展開サイトでは、1人または少数の管理者ですべてを管理し、委任管理者は存在しません。この場合、組み込みの [すべての管理権限を実行できる管理者] 役割（および [すべて] スコープ）の管理者を作成します。
- より多くのマシン、アプリケーション、およびデスクトップがあるサイトでは、委任管理者の配置が必要になります。何人かの管理者に、より専門的な管理責任（役割）を付与できます。たとえば、2人の「すべての管理権限を実行できる管理者」を設定して、残りをヘルプデスク管理者にします。さらに、マシンカタログなど、既定グループ（スコープ）のオブジェクトの管理を1人の管理者に委任することもできます。この場合、新しいスコープを作成して、組み込みの役割とそのスコープをペアにした管理者を作成します。
- 大規模サイトにおいても、より多くの（またはより詳細な）スコープと、特殊な役割を持つさまざまな管理者が必要になることがあります。この場合は、追加のスコープを作成または編集して、カスタムの役割を作成し、組み込みまたはカスタムの役割と既存または新しいスコープを持つ各管理者を作成します。

新しいスコープは、管理者を作成するときに作成できます。また、マシンカタログやホスト接続を作成または編集するときにスコープを指定することもできます。

## 管理者の作成と管理

ローカルの管理者アカウントを使用してサイトを作成するときは、すべてのオブジェクトに対する完全な管理権限を持つ管理者としてそのアカウントが設定されます。ただし、サイトを作成した後では、ローカル管理者には特別な特権は与えられません。

すべての管理タスクの実行権限を持つ管理者には、常に [すべて] のスコープが割り当てられます。これを変更することはできません。

デフォルトでは、管理者は有効になります。新しい管理者を作成するときに、その管理者が実際に作業を始めるまで管理者を無効にしておく必要が生じる場合があります。また、オブジェクトやスコープを再構成するときに、既存の管理者を一時的に無効にすることもできます。完全な管理権限を持つ管理者が1人しかいない環境では、その管理者を無効にすることはできません。管理者の有効/無効は、管理者を作成、コピー、または編集するときの [管理者を有効にする] チェックボックスで設定できます。

管理者を編集したりコピーしたりするときのダイアログボックスでスコープ/役割ペアを削除すると、その管理者とスコープ/役割ペアとの関連付けが削除され、個々のスコープや役割は削除されません。また、同じスコープ/役割ペアが割り当てられている管理者がいる場合でも、その関連付けは削除されません。

管理者を管理するには、Studioのナビゲーションペインで [構成] の [管理者] をクリックし、中央ペインの上部の [管理者] タブをクリックします。

- 管理者を作成するには、[操作] ペインの [管理者の作成] をクリックします。ユーザーアカウント名を入力するか参照し、スコープを選択または作成して、役割を選択します。新しい管理者はデフォルトで有効になりますが、無効にすることもできます。
- 管理者をコピーするには、中央ペインで管理者を選択し、[操作] ペインの [管理者のコピー] をクリックします。ユーザーアカウント名を入力するか参照します。必要に応じて、スコープ/役割ペアを編集または削除したり、新しいペアを追加したりできます。新しい管理者はデフォルトで有効になりますが、無効にすることもできます。
- 管理者を編集するには、中央ペインで管理者を選択し、[操作] ペインの [管理者の編集] をクリックします。必要に応じて、スコープ/役割ペアを編集または削除したり、新しいペアを追加したりできます。
- 管理者を削除するには、中央ペインで管理者を選択し、[操作] ペインの [管理者の削除] をクリックします。完全な管理権限を持つ管理者が1人しかいない環境では、その管理者を削除することはできません。

## 役割の作成と管理

役割には、64文字までのUnicode文字で名前を付けることができます。ただし、バックslash (\)、slash (/)、セミコロン (;)、コロン (:)、番号記号 (#)、コンマ (,)、アスタリスク (\*)、疑問符 (?)、等号 (=)、小なり記号 (<)、大なり記号 (>)、パイプ (|)、角かっこ ([ ])、丸かっこ ( ( ) )、二重引用符 (" )、およびアポストロフィ ( ' ) は使用できません。説明には、256文字までのUnicode文字を入力できます。

組み込みの役割を編集または削除することはできません。いずれかの管理者が使用しているカスタムの役割は削除できません。

注：カスタムの役割を作成するには、特定の製品エディションが必要です。カスタムの役割をサポートしないエディションでは、[操作] ペインに関連エントリが表示されません。

役割を管理するには、Studioのナビゲーションペインで [構成] の [役割] をクリックし、中央ペインの上部の [役割] タブをクリックします。

- 役割の詳細を表示するには、中央ペインでその役割を選択します。中央ペインの下部に、その役割のオブジェクトの種類および許可される権限が表示されます。ここ

で [管理者] タブをクリックすると、その役割が割り当てられている管理者が表示されます。

- カスタムの役割を作成するには、[操作] ペインの [役割の作成] をクリックします。名前と説明を入力します。この役割に割り当てるオブジェクトの種類と権限を選択します。
- 役割をコピーするには、中央ペインで役割を選択し、[操作] ペインの [役割のコピー] をクリックします。必要に応じて、役割の名前、説明、および権限を変更します。
- カスタムの役割を編集するには、中央ペインで役割を選択し、[操作] ペインの [役割の編集] をクリックします。必要に応じて、役割の名前、説明、および権限を変更します。
- カスタムの役割を削除するには、中央ペインで役割を選択し、[操作] ペインの [役割の削除] をクリックします。確認のメッセージが表示されたら、[削除] をクリックします。

## スコープの作成と管理

サイトを作成すると、[すべて] のスコープが使用可能になります。このスコープは削除できません。

スコープを作成するには、次の手順を使用します。管理者を作成するときにスコープを作成することもできます。すべての管理者は、少なくとも1つの役割とスコープのペアが割り当てられている必要があります。デスクトップ、マシンカタログ、アプリケーション、またはホストを作成したり編集したりするときに、それらを既存のスコープに追加できます。ただし、特定のスコープに追加しない場合でも、自動的に [すべて] のスコープに追加されます。

サイトの作成および委任管理オブジェクト (スコープおよび役割) をスコープに含めることはできません。ただし、スコープに含めることができないオブジェクトも [すべて] のスコープには含まれています。すべての管理タスクの実行権限を持つ管理者には、常に [すべて] のスコープが割り当てられます。マシン、電源操作、デスクトップ、およびセッションはスコープに含まれません。これらのオブジェクトに対する管理者は、マシンカタログまたはデリバリーグループで割り当てることができます。

スコープには、64文字までのUnicode文字で名前を付けることができます。ただし、バックスラッシュ (\)、スラッシュ (/)、セミコロン (;)、コロン (:)、番号記号 (#)、コンマ (,)、アスタリスク (\*)、疑問符 (?)、等号 (=)、小なり記号 (<)、大なり記号 (>)、パイプ (|)、角かっこ ([ ])、丸かっこ ( ( ) )、二重引用符 (")、およびアポストロフィ (') は使用できません。説明には、256文字までのUnicode文字を入力できます。

スコープをコピーまたは編集するときにオブジェクトをスコープから削除すると、管理者がそのオブジェクトにアクセスできなくなる可能性があることに注意してください。編集するスコープにいくつかの役割が関連付けられている場合は、編集によりスコープと役割のペアが使用できなくなるかどうかを確認してください。

スコープを管理するには、Studioのナビゲーションペインで [構成] の [管理者] をクリックし、中央ペインの上部の [スコープ] タブをクリックします。

- スコープを作成するには、[操作] ペインの [スコープの作成] をクリックします。名前と説明を入力します。オブジェクトの種類 ( [デリバリーグループ] チェックボックスなど) を選択すると、その種類のすべてのオブジェクトがスコープに追加されます。特定のオブジェクトを追加するには、オブジェクトの種類を開き、個々のオブジェクトを選択します (営業部で使用される特定のデリバリーグループを選択する場合など)。
- スコープをコピーするには、中央ペインでスコープを選択し、[操作] ペインの [スコープのコピー] をクリックします。名前と説明を入力します。必要に応じて、オブジェクトの種類とオブジェクトを変更します。
- スコープを編集するには、中央ペインでスコープを選択し、[操作] ペインの [スコープの編集] をクリックします。必要に応じて、名前、説明、オブジェクトの種類、およびオブジェクトを変更します。
- スコープを削除するには、中央ペインでスコープを選択し、[操作] ペインの [スコープの削除] をクリックします。確認のメッセージが表示されたら、[削除] をクリックします。

## レポートの作成

次の2種類の委任管理レポートを作成できます。

- 管理者に関連付けられているスコープ/役割ペアと各種類のオブジェクト (デリバリーグループ、マシンカタログなど) に対する個々の権限の一覧についてのHTMLレポート。Studioで生成できます。

このレポートを作成するには、ナビゲーションペインで [構成] の [管理者] を選択します。中央ペインで管理者を選択し、操作ペインで [レポートの作成] をクリックします。

このレポートは、管理者の作成、コピー、および編集時に作成することもできます。

- 組み込みおよびカスタムの役割とそれらに関連付けられた権限を一覧表示するHTMLまたはCSVレポート。このレポートは、PowerShellスクリプト OutputPermissionMapping.ps1を実行して生成します。このスクリプトを実行するには、すべての管理権限を実行できる管理者、読み取り専用管理者、または役割の読み取り権限を持つ管理者である必要があります。このスクリプトは、Program Files\Citrix\DelegatedAdmin\SnapIn\Citrix.DelegatedAdmin.Admin.V1\Scripts\にインストールされています。

構文：

```
OutputPermissionMapping.ps1 [-Help] [-Csv] [-Path <string>] [-AdminAddress <string>] [-Show] [<CommonParameters>]
```

| パラメーター         | 説明                        |
|----------------|---------------------------|
| -Help          | スクリプトのヘルプを表示します。          |
| -Csv           | CSVレポートを作成します。デフォルト値：HTML |
| -Path <string> | 出力先を指定します。デフォルト値：stdout   |

|                         |                                                                                                                                 |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| PathAddress<br><string> | 説明先のDelivery ControllerのIPアドレスまたはホスト名を指定します。デフォルト値 : localhost                                                                  |
| -Show                   | (-Path/パラメーターを指定した場合のみ有効) ファイルに出力する場合に-Showを指定すると、レポートが適切なアプリケーションプログラム (Webブラウザなど) で表示されます。                                    |
|                         | Verbose、Debug、ErrorAction、ErrorVariable、WarningAction、WarningVariable、OutBuffer、およびOutVariable。詳しくは、Microsoft社のドキュメントを参照してください。 |

次の例では、Roles.htmlという名前のファイルにHTMLテーブルが出力され、Webブラウザで表示されます。

```
& "$env:ProgramFiles\Citrix\DelegatedAdmin\SnapIn\Citrix.DelegatedAdmin.Admin.V1\Scripts\OutputPermissionMapping.ps1" -Path Roles.html -Show
```

次の例では、Roles.csvという名前のファイルにCSVテーブルが出力されます。このテーブルは自動的に表示されません。

```
& "$env:ProgramFiles\Citrix\DelegatedAdmin\SnapIn\Citrix.DelegatedAdmin.Admin.V1\Scripts\OutputPermissionMapping.ps1" -CSV -Path Roles.csv
```

上の例をWindowsコマンドプロンプトから実行する場合は、次のコマンドを実行します。

```
powershell -command "& '%ProgramFiles%\Citrix\DelegatedAdmin\SnapIn\Citrix.DelegatedAdmin.Admin.V1\Scripts\OutputPermissionMapping.ps1' -CSV -Path Roles.csv"
```



# スマートカード

Jan 04, 2017

スマートカードおよび同等のテクノロジーは、このアークティクルに記載されているガイドライン内でサポートされています。XenAppまたはXenDesktopでスマートカードを使用するには、以下を行う必要があります。

- 所属する組織における、スマートカードの使用に関するセキュリティポリシーを理解します。たとえば、スマートカードがどのように発行され、ユーザーがそれをどのように保護するかについてこれらのポリシーで規定してあることがあります。XenAppおよびXenDesktopの環境では、これらのポリシーの一部の変更が必要になる場合があります。
- どのユーザーデバイスの種類、オペレーティングシステム、および公開アプリケーションがスマートカードとともに使用されるかを決定します。
- スマートカードテクノロジー全般および選択したスマートカードベンダーのハードウェアとソフトウェアについて理解します。
- 分散環境でのデジタル証明書の展開管理方法について理解します。

## スマートカードの種類

エンタープライズ向けとコンシューマー向けのスマートカードは、寸法も電気コネクタも同じで、同じスマートカードリーダーを使用できます。

エンタープライズ向けのスマートカードにはデジタル証明書が含まれています。これらのスマートカードはWindowsログオンをサポートしていて、ドキュメントや電子メールのデジタル署名と暗号化のためのアプリケーションと連携して使用できます。XenAppとXenDesktopは、これらの用途をサポートしています。

コンシューマー向けのスマートカードにはデジタル証明書は含まれていませんが、共有シークレットが含まれています。これらのスマートカードは、支払い（チップと署名、チップとPINクレジットカードなど）をサポートできます。これらのスマートカードは、Windowsログインや一般的なWindowsアプリケーションをサポートしていません。これらのスマートカードと合わせて使用するには、特別なWindowsアプリケーションと、適切なソフトウェアインフラストラクチャ（支払いカードネットワークへの接続など）が必要です。XenAppまたはXenDesktopでのこのような特別なアプリケーションのサポートについて詳しくは、Citrix担当者にお問い合わせください。

エンタープライズ向けスマートカードには、互換性のある同等のものが存在し、類似した方法で使用できます。

- スマートカードと同等のUSBトークンはUSBポートに直接接続します。これらのUSBトークンは通常USBフラッシュドライブのサイズですが、携帯電話で使用されるSIMカードと同じくらい小さいものもあります。それらは、スマートカードとUSBスマートカードリーダーの組み合わせとして表示されます。
- Windowsトラステッドプラットフォームモジュール (TPM) を使用する仮想スマートカードは、スマートカードとして表示されます。これらの仮想スマートカードは、Citrix Receiver 4.3以上を使用して、Windows 8およびWindows 10でサポートされます。
  - XenAppおよびXenDesktopの7.6 FP3よりも前のバージョンは、仮想スマートカードをサポートしていません。
  - 仮想スマートカードについて詳しくは、「[Virtual Smart Card Overview](#)」を参照してください。

注：「仮想スマートカード」という用語は、単にユーザーコンピューターに保存されたデジタル証明書についても使用されます。これらのデジタル証明書は、厳密にはスマートカードと同等ではありません。

XenAppおよびXenDesktopのスマートカードのサポートは、MicrosoftのPC/SC (Personal Computer/Smart Card) 標準仕様に基づいています。スマートカードおよびスマートカードデバイスは、使用するWindowsオペレーティングシステムでサ

ポートされており、Microsoft WHQL (Windows Hardware Quality Lab) により承認されている必要があります。PC/SCに準拠しているハードウェアについては、Microsoft社のドキュメントを参照してください。その他のタイプのユーザーデバイスは、PS/SC標準に準拠していることがあります。詳しくは、<http://www.citrix.com/ready/>でCitrix Readyプログラムについて参照してください。

通常、各ベンダーのスマートカードまたは同等のものには、別々のデバイスドライバーが必要です。ただし、スマートカードがNIST Personal Identity Verification (PIV) 標準などの標準に準拠している場合、一定範囲のスマートカードに単一のデバイスドライバーを使用できる場合があります。デバイスドライバーをユーザーデバイスとVirtual Delivery Agent (VDA) の両方にインストールする必要があります。多くの場合、デバイスドライバーはCitrixパートナーから入手可能なスマートカードミドルウェアパッケージの一部として提供されます。スマートカードミドルウェアパッケージにより、高度な機能が提供されます。デバイスドライバーは、暗号化サービスプロバイダー (CSP)、キーストレージプロバイダー (KSP)、ミニドライバーとして説明されることもあります。

Windowsシステムでは、以下のスマートカードとミドルウェアでの動作確認が行われています。ただし、そのほかのスマートカードおよびミドルウェアも使用できます。Citrix互換のスマートカードとミドルウェアについて詳しくは、<http://www.citrix.com/ready/ja>を参照してください。

| ミドルウェア                           | スマートカード          |
|----------------------------------|------------------|
| ActivClient 7.0 (DoDモード有効)       | DoD CACカード       |
| PIVモードのActivClient 7.0           | NIST PIVカード      |
| Microsoft ミニドライバー                | NIST PIVカード      |
| GemAlto Mini Driver for .NET カード | GemAlto .NET v2+ |
| Microsoft ネイティブドライバー             | 仮想スマートカード (TPM)  |

他の種類のデバイスでのスマートカード使用法について詳しくは、そのデバイスに関するCitrix Receiverのドキュメントを参照してください。XenDesktopによるPIV使用法について詳しくは、『[Configuring Citrix XenDesktop 7.6 and NetScaler Gateway 10.5 with PIV Smart Card Authentication](#)』を参照してください。

他の種類のデバイスでのスマートカード使用法について詳しくは、そのデバイスに関するCitrix Receiverのドキュメントを参照してください。

## リモートPCアクセス

オフィスで動作する、物理的なWindows 10、Windows 8、またはWindows 7マシンにリモートアクセスする場合は、スマートカードがサポートされます。Windows XPマシンへのリモートアクセスでは、スマートカードはサポートされません。

以下のスマートカードが、リモートPCアクセス機能でテストされています。

| ミドルウェア               | スマートカード          |
|----------------------|------------------|
| Gemalto .NET ミニドライバー | Gemalto .NET v2+ |

|                               |           |
|-------------------------------|-----------|
| ActivIdentity ActivClient 6.2 | NIST PIV  |
| ActivIdentity ActivClient 6.2 | CAC       |
| Microsoft ミニドライバ              | NIST PIV  |
| Microsoft ネイティブドライバ           | 仮想スマートカード |

## スマートカードリーダーの種類

スマートカードリーダーはユーザーデバイス内に作成されることもありますし、別にユーザーデバイスに（通常はUSBまたはBluetoothで）接続することもあります。USB Chip/Smart Card Interface Devices (CCID) 仕様に準拠する接触カードリーダーがサポートされます。これらのカードリーダーでは、ユーザーがスマートカードをスロットに挿入したりスワイプしたりします。Deutsche Kreditwirtschaft (DK) 標準は、接触カードリーダーの4つのクラスを定義しています。

- Class 1スマートカードリーダーは最も一般的で、通常1つのみのスロットを備えています。Class 1スマートカードリーダーは通常、オペレーティングシステム付属の標準CCIDデバイスドライバでサポートされます。
- Class 2スマートカードリーダーには、ユーザーデバイスがアクセスできない安全なキーパッドも含まれています。Class 2スマートカードリーダーは、内蔵の安全なキーパッドがあるキーボードに搭載される場合があります。Class 2スマートカードリーダーについては、Citrixの担当者に連絡してください。安全なキーパッドの機能を有効化するには、リーダー固有のデバイスドライバが必要になる場合があります。
- Class 3スマートカードリーダーには、安全なディスプレイも含まれます。Class 3スマートカードリーダーはサポートされません。
- Class 4スマートカードリーダーには、安全なトランザクションモジュールも含まれます。Class 4スマートカードリーダーはサポートされません。

注：スマートカードリーダーのクラスは、USBデバイスのクラスには無関係です。

スマートカードリーダーは、対応するデバイスドライバとともにユーザーデバイスにインストールする必要があります。

## ユーザーエクスペリエンス

スマートカードのサポートは、デフォルトで有効な特定のICA/HDXスマートカード仮想チャネルを使用して、XenAppおよびXenDesktopに統合されています。

**重要：**スマートカードリーダーでは一般のUSBリダイレクションを使用しないでください。一部のスマートカードリーダーはこれはデフォルトで無効にされており、有効化した場合サポートされなくなります。

同一ユーザーデバイス上で、複数のスマートカードやスマートカードリーダーを使用することは可能ですが、パススルー認証を使用する場合は1枚のスマートカードを挿入した状態で仮想デスクトップまたはアプリケーションを開始する必要があります。アプリケーション内でスマートカードを使用する場合（デジタル署名または暗号化機能など）、スマートカードの挿入またはPINの入力を求めるメッセージが表示されることがあります。これは、同時に複数のスマートカードが挿入されている場

合に発生します。

- 適切なスマートカードを挿入しているにもかかわらずスマートカードの挿入を求めるメッセージが表示された場合は、[キャンセル] をクリックするよう通知します。
- ただし、PINの入力が求められた場合は、PINを再入力する必要があります。

Microsoft社のベーススマートカード暗号化サービスプロバイダー（CSP）によるスマートカードを使用する場合、Windows Server 2008または2008 R2が動作するサーバー上のアプリケーションにユーザーがアクセスすると、ほかのユーザーがスマートカードでログオンできなくなります。これについての詳細および修正プログラムについては、<http://support.microsoft.com/kb/949538>を参照してください。

カード管理システムまたはベンダーのユーティリティを使ってPINをリセットできます。

## スマートカードを展開する前の確認事項

- スマートカードリーダーのデバイスドライバーを入手して、ユーザーデバイスにインストールする必要があります。Microsoftにより提供されるCCIDデバイスドライバーは、多くのスマートカードリーダーで使用できます。
- スマートカードベンダーからデバイスドライバーと暗号化サービスプロバイダー（CSP）ソフトウェアを入手して、ユーザーデバイスと仮想デスクトップの両方にインストールします。このドライバーとCSPソフトウェアは、XenAppやXenDesktopと互換性がある必要があります。詳しくは、ベンダーのドキュメントを参照してください。ミニドライバーモデルのスマートカードを使用する仮想デスクトップでは、スマートカードミニドライバーが自動的にダウンロードされます。また、<http://catalog.update.microsoft.com>またはベンダーから入手することもできます。さらに、PKCS#11ミドルウェアが必要な場合は、カードベンダーから入手してください。
- **重要**：Citrixソフトウェアをインストールする前に、物理的なコンピューターにドライバーとCSPソフトウェアをインストールしてテストすることをお勧めします。
- Windows 10で実行するInternet Explorerでスマートカードを実行するユーザーの信頼済みサイトの一覧にCitrix Receiver for Web URLを追加します。Windows 10では、Internet Explorerは信頼済みサイトのデフォルトで保護モードでは実行しません。
- PKI（Public Key Infrastructure：公開キー基盤）が適切に構成されていることを確認します。つまり、アカウントマッピングのための証明書がActive Directory環境に対して正しく構成されており、ユーザー証明書の検証を正しく実行できることを確認します。
- Citrix ReceiverやStoreFrontなど、スマートカードで使用するほかのCitrixコンポーネントのシステム要件を満たしていることを確認します。
- サイト内の以下のサーバーにアクセスできることを確認します。
  - スマートカード上のログオン証明書に関連付けられているユーザーアカウント用のActive Directoryドメインコントローラー
  - Delivery Controller
  - Citrix StoreFront
  - Citrix NetScaler Gateway/Citrix Access Gateway 10.x
  - VDA
  - Microsoft Exchange Server（リモートPCアクセスの場合はオプション）

## スマートカード使用の有効化

手順1：カードの発行ポリシーに従って、ユーザーにスマートカードを発行します。

**手順2**： 必要に応じて、ユーザーがリモートPCアクセスを実行できるようにスマートカードをセットアップします。

**手順3**： Delivery ControllerとStoreFrontをインストールして（未インストールの場合）、スマートカードのリモート処理用に構成します。

**手順4**： StoreFrontで、スマートカードの使用を有効にします。詳しくは、StoreFrontドキュメントの「スマートカード認証の構成」を参照してください。

**手順5**： NetScaler Gateway/Access Gatewayで、スマートカードの使用を有効にします。詳しくは、NetScalerドキュメントの「認証と承認の構成」および「Web Interfaceでのスマートカードアクセスの構成」を参照してください。

**手順6**： VDAsで、スマートカードの使用を有効にします。

- VDAに必要なアプリケーションおよび更新がインストール済みであることを確認します。
- ミドルウェアをインストールします。
- ユーザーデバイス上のCitrix Receiverと仮想デスクトップセッション間でスマートカードデータ通信が行われるように、スマートカードのリモート処理をセットアップします。

**手順7**： ユーザーデバイス（ドメインに属しているマシンと属していないマシンを含む）でスマートカードの使用を有効にします。詳しくは、StoreFrontドキュメントの「スマートカード認証の構成」を参照してください。

- 証明機関のルート証明書とその証明機関の証明書をデバイスのキーストア内にインポートします。
- ベンダーが提供するスマートカードミドルウェアをインストールします。
- Citrix Receiver for Windowsをインストールおよび構成して、グループポリシー管理コンソールを使ってicaclient.admをインポートします。また、スマートカード認証を有効にします。

**手順8**： 展開をテストします。テストユーザーのスマートカードで仮想デスクトップを起動して、展開が正しく構成されていることを確認します。すべてのアクセス方法（たとえば、Internet ExplorerおよびCitrix Receiverを介したデスクトップアクセスなど）をテストします。

# スマートカード展開

Jan 04, 2017

この製品バージョンおよびこのバージョンと以前のバージョンとの混在環境では、以下の種類のスマートカード展開がサポートされます。そのほかの構成でも使用できる場合がありますが、サポートの対象外です。

| 種類                                                             | StoreFrontへの接続          |
|----------------------------------------------------------------|-------------------------|
| ローカルのドメイン参加コンピューター                                             | 直接接続                    |
| ドメイン参加コンピューターからのリモートアクセス                                       | NetScaler Gateway経由の接続  |
| ドメイン不参加コンピューター                                                 | 直接接続                    |
| ドメイン不参加コンピューターからのリモートアクセス                                      | NetScaler Gateway経由の接続  |
| デスクトップアプライアンスサイトにアクセスするドメイン不参加コンピューターおよびシンクライアント               | デスクトップアプライアンスサイト経由の接続   |
| XenApp Servicesサイト経由でStoreFrontにアクセスするドメイン参加コンピューターおよびシンクライアント | XenApp Servicesサイト経由の接続 |

展開の種類は、スマートカードリーダーが接続されているユーザーデバイスの特徴により定義されます。

- デバイスがドメインに参加しているか参加していないか。
- デバイスがStoreFrontにどのように接続するか。
- 仮想デスクトップやアプリケーションの表示にどのソフトウェアを使用するか。

これらの展開では、Microsoft WordやMicrosoft Excelなど、スマートカード対応のアプリケーションを使用できます。ユーザーは、これらのアプリケーションを使用してドキュメントにデジタル署名を追加したり、ドキュメントを暗号化したりできます。

## 2モード認証

これらの各展開で可能な箇所では、スマートカードを使用するのか、ユーザー名およびパスワードを入力するのかをユーザーに選択させる2モード認証をReceiverがサポートします。この機能は、ユーザーがスマートカードを使用できない場合（スマートカードを自宅に忘れた場合や資格情報の有効期限が切れた場合など）に便利です。

ドメイン不参加デバイスのユーザーはReceiver for Windowsに直接ログオンするため、管理者は指定ユーザー認証へのフォールバックを有効にすることができます。2モード認証を構成を構成した場合、ユーザーは最初にスマートカードとPINを使ったログオンを要求されますが、スマートカードでログオンできない場合は指定ユーザー認証を選択することができます。

NetScaler Gatewayを使用する環境では、ユーザーはデバイスにログオンし、NetScaler Gatewayの認証を受けるようにReceiver for Windowsから要求されます。これはドメイン参加デバイスとドメイン不参加デバイスの両方に適用されます。ユーザーは、スマートカードとPINを使って、または指定ユーザーの資格情報を使ってNetScaler Gatewayにログオンできます。これにより、NetScaler Gatewayにログオンするときの2モード認証をユーザーに提供できます。ユーザーがStoreFrontに透過的に認証されるように、NetScaler GatewayからStoreFrontへのパススルー認証を構成し、スマートカードユーザーの資格情報の検証をNetScaler Gatewayに委任します。

## 複数Active Directoryフォレストでの考慮事項

Citrix環境では、スマートカードは単一のフォレスト内でサポートされます。フォレスト間でのスマートカード認証には、すべてのユーザーアカウントに対する直接の双方向の信頼関係が必要です。より複雑なマルチフォレスト展開（一方向のみまたはそのほかの信頼関係が設定された複数フォレスト展開）はサポートされていません。

リモートデスクトップを含むCitrix環境でスマートカードを使用できます。この機能は、（スマートカードが接続されるユーザーデバイス上に）ローカルにインストールしたり、（ユーザーデバイスが接続するリモートデスクトップ上に）リモートでインストールしたりできます。

### スマートカード取り出し時の動作ポリシー

スマートカード取り出し時の動作ポリシーの設定により、セッション中にスマートカードリーダーからカードを取り出したときの処理が制御されます。このポリシーは、Windowsオペレーティングシステムで設定します。

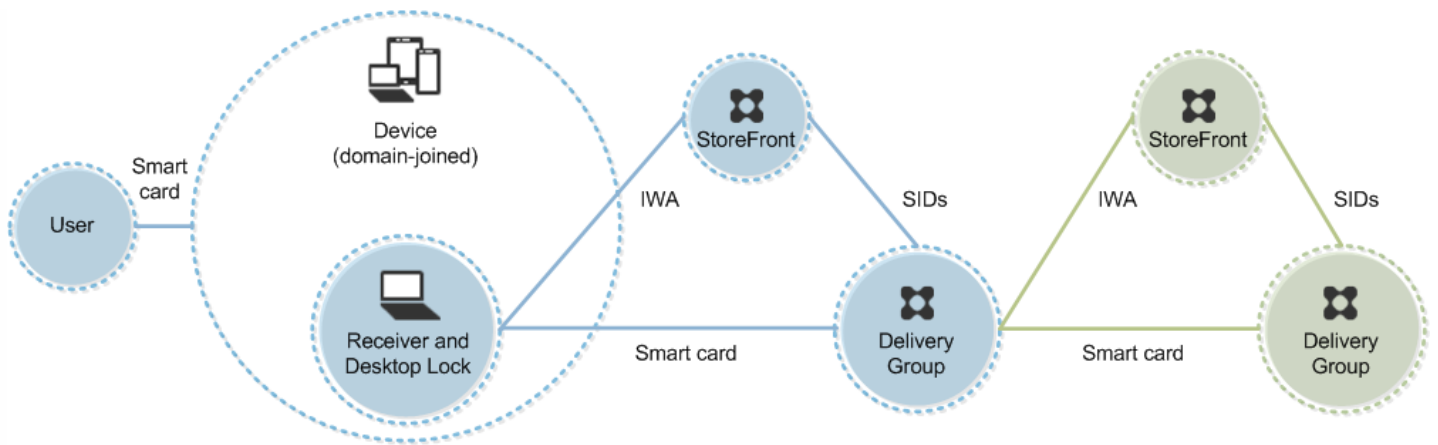
| ポリシー設定                   | デスクトップの動作                                                              |
|--------------------------|------------------------------------------------------------------------|
| 何もしない                    | 何もしません。                                                                |
| ワークステーションをロック            | デスクトップセッションは切断され、仮想デスクトップはロックされます。                                     |
| ログオフを強制する                | ユーザーは強制的にログオフされます。ネットワーク接続が失われ、この設定が有効な場合、セッションはログオフされてユーザーのデータは消失します。 |
| リモートターミナルサービスセッションの場合に切断 | セッションは切断され、仮想デスクトップはロックされます。                                           |

### 証明書失効のチェック

証明書失効のチェックが有効な場合、スマートカードの証明書が有効かどうかを検出されます。証明書が無効な場合、ユーザー認証に失敗したり、その証明書に関連付けられているデスクトップやアプリケーションへのアクセスが拒否されたりします。たとえば、メールの復号化用の証明書が無効な場合、暗号化されたメールを復号化できなくなります。同じスマートカード上に有効なほかの証明書がある場合、その機能については有効なままとなります。たとえば、認証用の証明書が有効な場合、ユーザー認証に成功します。

### 展開例：ドメイン参加コンピューター

この展開には、Desktop Viewerを実行し、StoreFrontに直接接続する、ドメインに参加しているユーザーデバイスが含まれています。

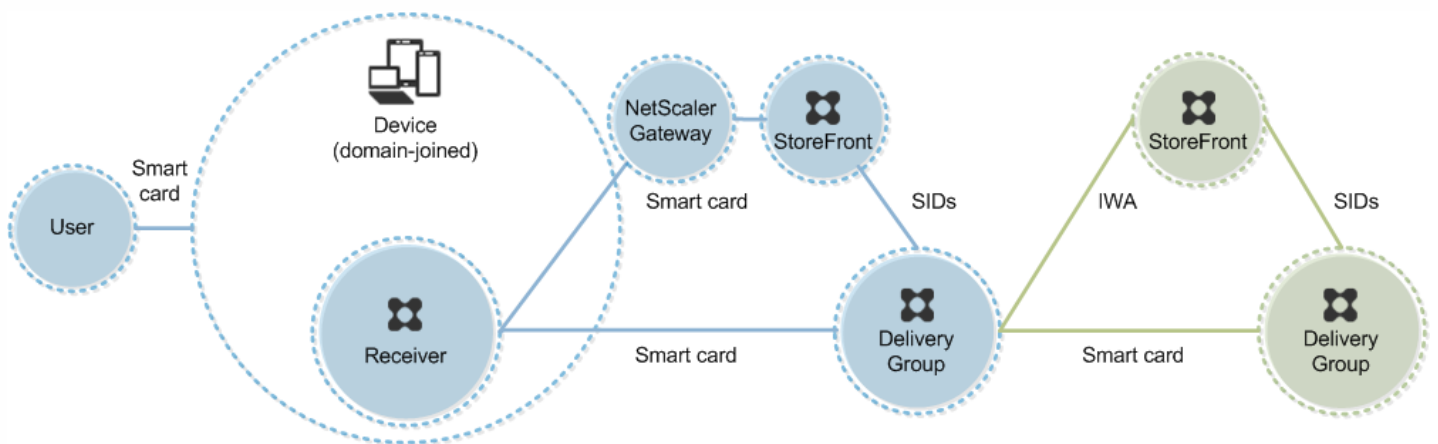


ユーザーは、スマートカードとPINを使ってデバイスにログオンします。Receiverは、StoreFrontサーバーにアクセスするユーザーを統合Windows認証（IWA）で認証します。StoreFrontにより、ユーザーのセキュリティ識別子（SID）がXenAppまたはXenDesktopに渡されます。Receiverでシングルサインオン機能が構成されているため、ユーザーが仮想デスクトップやアプリケーションを起動するときにPINを再入力する必要はありません。

この展開は、2つ目のStoreFrontサーバーとアプリケーションをホストするサーバーを追加してダブルホップ形式に拡張できます。仮想デスクトップのReceiverは、2つ目のStoreFrontサーバーへの認証を実行します。この2つ目の接続では任意の認証方法を使用できます。最初の接続で使用した認証方法を2つ目の接続で再使用したり、または2つ目の接続で異なる方法を使用したりできます。

#### 展開例：ドメイン参加コンピューターからのリモートアクセス

この展開には、Desktop Viewerを実行し、NetScaler Gateway/Access Gatewayを介してStoreFrontに接続する、ドメインに参加しているユーザーデバイスが含まれています。



ユーザーはスマートカードとPINを使ってデバイスにログオンし、次にNetScaler Gateway/Access Gatewayにもう一度ログオンします。この展開ではReceiverで2モード認証を使用できるため、この2つ目のログオンではスマートカードとPINを使用したりユーザー名とパスワードを入力したりできます。

ユーザーは自動的にStoreFrontにログオンし、ユーザーセキュリティ識別子（SID）がXenAppまたはXenDesktopに渡されます。Receiverでシングルサインオン機能が構成されているため、ユーザーが仮想デスクトップやアプリケーションを起動するときにPINを再入力する必要はありません。

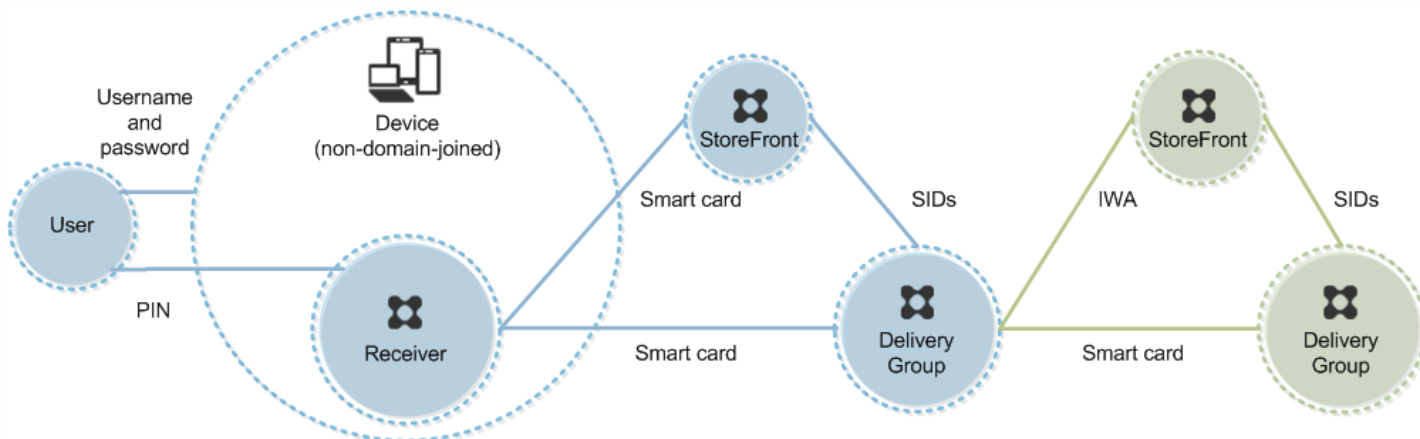
この展開は、2つ目のStoreFrontサーバーとアプリケーションをホストするサーバーを追加してダブルホップ形式に拡張でき



ます。仮想デスクトップのReceiverは、2つ目のStoreFrontサーバーへの認証を実行します。この2つ目の接続では任意の認証方法を使用できます。最初の接続で使用した認証方法を2つ目の接続で再使用したり、または2つ目の接続で異なる方法を使用したりできます。

#### 展開例：ドメイン不参加コンピューター

この展開には、Desktop Viewerを実行し、StoreFrontに直接接続する、ドメイン不参加のユーザーデバイスが含まれています。



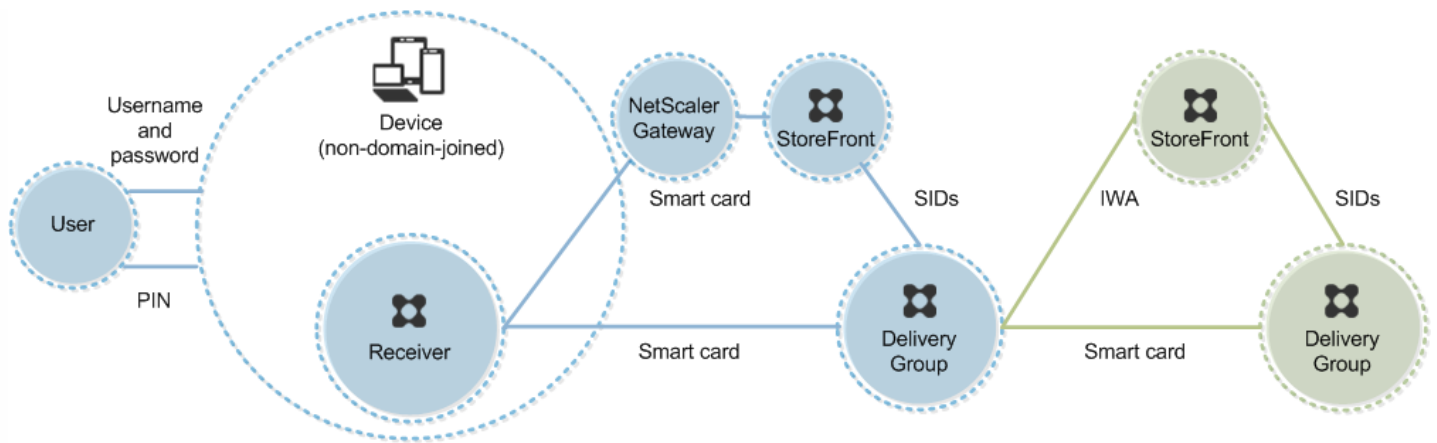
ユーザーがデバイスにログオンします。通常はユーザー名とパスワードを入力しますが、デバイスがドメインに参加していないため、このログオンでの資格情報の入力はありません。この展開では2モード認証を使用できるため、ReceiverではスマートカードとPIN、またはユーザー名とパスワードのいずれかの入力が必要です。その後、ReceiverがStorefrontへの認証を実行します。

StoreFrontにより、ユーザーのセキュリティ識別子 (SID) がXenAppまたはXenDesktopに渡されます。この展開ではシングルサインオン機能を使用できないため、ユーザーが仮想デスクトップやアプリケーションを起動するときにPINを再入力する必要があります。

この展開は、2つ目のStoreFrontサーバーとアプリケーションをホストするサーバーを追加してダブルホップ形式に拡張できます。仮想デスクトップのReceiverは、2つ目のStoreFrontサーバーへの認証を実行します。この2つ目の接続では任意の認証方法を使用できます。最初の接続で使用した認証方法を2つ目の接続で再使用したり、または2つ目の接続で異なる方法を使用したりできます。

#### 展開例：ドメイン不参加コンピューターからのリモートアクセス

この展開には、Desktop Viewerを実行し、StoreFrontに直接接続する、ドメイン不参加のユーザーデバイスが含まれています。



ユーザーがデバイスにログオンします。通常はユーザー名とパスワードを入力しますが、デバイスがドメインに参加していないため、このログオンでの資格情報の入力には必須ではありません。この展開では2モード認証を使用できるため、ReceiverではスマートカードとPIN、またはユーザー名とパスワードのいずれかの入力が必要とされます。その後、ReceiverがStorefrontへの認証を実行します。

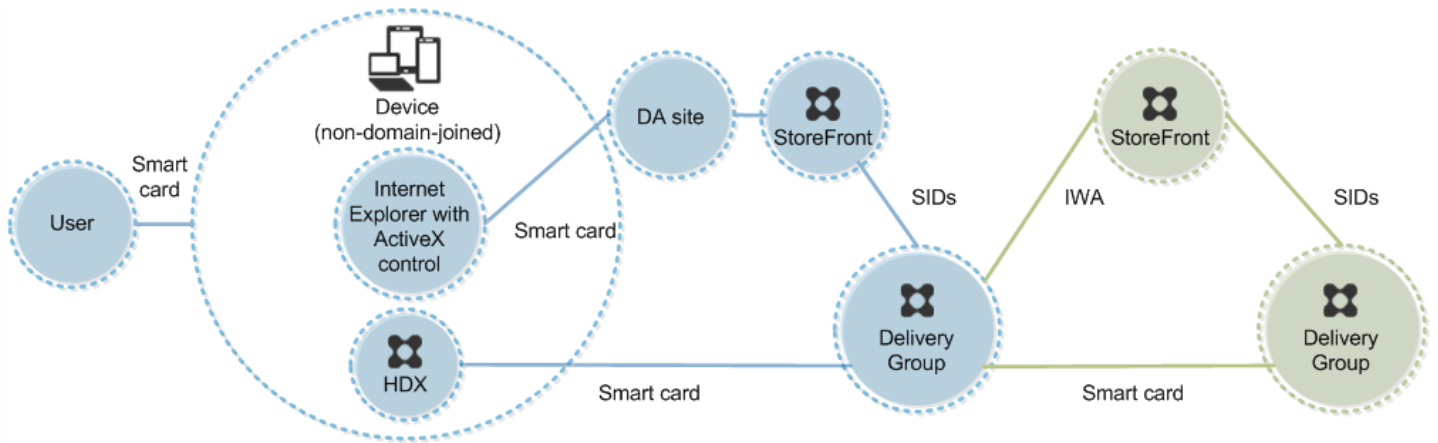
StoreFrontにより、ユーザーのセキュリティ識別子 (SID) がXenAppまたはXenDesktopに渡されます。この展開ではシングルサインオン機能を使用できないため、ユーザーが仮想デスクトップやアプリケーションを起動するときにPINを再入力する必要があります。

この展開は、2つ目のStoreFrontサーバーとアプリケーションをホストするサーバーを追加してダブルホップ形式に拡張できます。仮想デスクトップのReceiverは、2つ目のStoreFrontサーバーへの認証を実行します。この2つ目の接続では任意の認証方法を使用できます。最初の接続で使用した認証方法を2つ目の接続で再使用したり、または2つ目の接続で異なる方法を使用したりできます。

**展開例：**デスクトップアプライアンスサイトにアクセスするドメイン不参加コンピューターおよびシンクライアント

この展開には、Desktop Lockを実行し、デスクトップアプライアンスサイトを介してStoreFrontに接続する、ドメイン不参加のユーザーデバイスが含まれています。

Desktop Lockは、XenApp、XenDesktop、およびCitrix VDI-in-a-Boxと一緒にリリースされる個別のコンポーネントです。Desktop Viewerの代替として使用でき、主に再目的化されたWindowsコンピューターおよびWindowsシンクライアント向けに設計されています。Desktop Lockはユーザーデバイス上のWindows Shellとタスクマネージャーを置き換えるもので、これによりユーザーはそのデバイスに直接アクセスできなくなります。Desktop Lockにより、ユーザーにはWindows ServerおよびWindows Desktopのデスクトップが提供されます。Desktop Lockのインストールは必須ではありません。



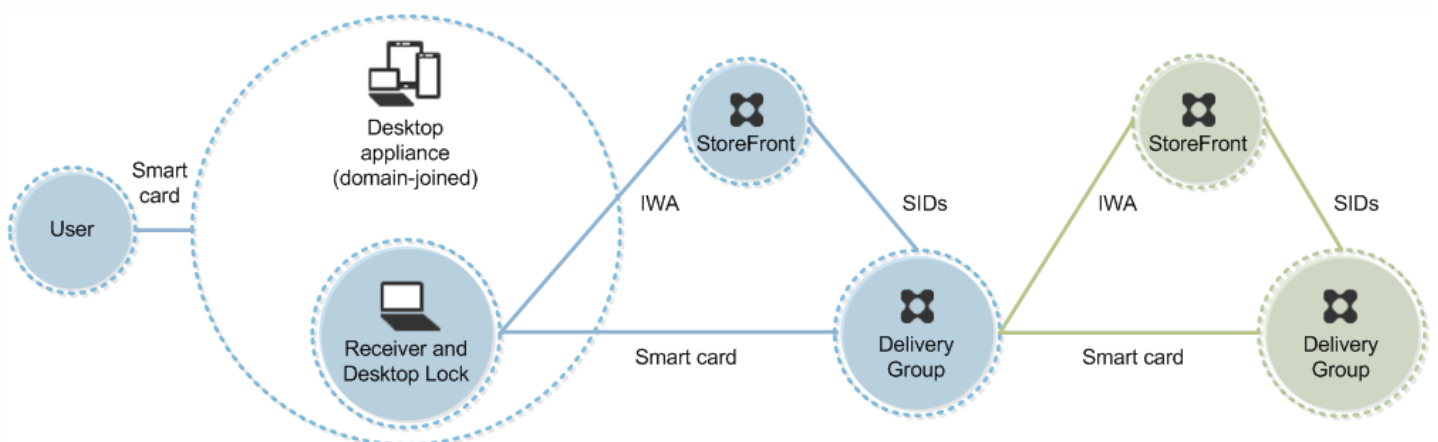
ユーザーは、スマートカードを使ってデバイスにログオンします。Desktop Lockを実行するデバイスは、キオスクモードで作るInternet Explorerを介してデスクトップアプライアンスサイトを起動するように構成されます。サイトのActiveXコントロールによりPINの入力が求められ、それがStoreFrontに送信されます。StoreFrontにより、ユーザーのセキュリティ識別子（SID）がXenAppまたはXenDesktopに渡されます。割り当てられたデスクトップグループ一覧で使用可能な（アルファベット順で）最初のデスクトップが起動します。

この展開は、2つ目のStoreFrontサーバーとアプリケーションをホストするサーバーを追加してダブルホップ形式に拡張できます。仮想デスクトップのReceiverは、2つ目のStoreFrontサーバーへの認証を実行します。この2つ目の接続では任意の認証方法を使用できます。最初の接続で使用した認証方法を2つ目の接続で再使用したり、または2つ目の接続で異なる方法を使用したりできます。

#### 展開例：XenApp Servicesサイト経由でStoreFrontにアクセスするドメイン参加コンピューターおよびシンクライアント

この展開には、Desktop Lockを実行し、XenApp Services URLを介してStoreFrontに接続する、ドメインに参加しているユーザーデバイスが含まれています。

Desktop Lockは、XenApp、XenDesktop、およびCitrix VDI-in-a-Boxと一緒にリリースされる個別のコンポーネントです。Desktop Viewerの代替として使用でき、主に再目的化されたWindowsコンピューターおよびWindowsシンクライアント向けに設計されています。Desktop Lockはユーザーデバイス上のWindows Shellとタスクマネージャーを置き換えるもので、これによりユーザーはそのデバイスに直接アクセスできなくなります。Desktop Lockにより、ユーザーにはWindows ServerおよびWindows Desktopのデスクトップが提供されます。Desktop Lockのインストールは必須ではありません。



ユーザーは、スマートカードとPINを使ってデバイスにログオンします。デバイス上でDesktop Lockが動作している場合は、

StoreFrontサーバーでのユーザー認証に統合Windows認証 (IWA) が使用されます。StoreFrontにより、ユーザーのセキュリティ識別子 (SID) がXenAppまたはXenDesktopに渡されます。Receiverでシングルサインオン機能が構成されているため、ユーザーが仮想デスクトップを起動するときにPINを再入力する必要はありません。

この展開は、2つ目のStoreFrontサーバーとアプリケーションをホストするサーバーを追加してダブルホップ形式に拡張できます。仮想デスクトップのReceiverは、2つ目のStoreFrontサーバーへの認証を実行します。この2つ目の接続では任意の認証方法を使用できます。最初の接続で使用した認証方法を2つ目の接続で再使用したり、または2つ目の接続で異なる方法を使用したりできます。

# スマートカードを使用したパススルー認証とシングルサインオン

Jan 04, 2017

## パススルー認証

仮想デスクトップへのスマートカードによるパススルー認証は、Windows 10、Windows 8、Windows 7 Service Pack 1 Enterpriseエディション、およびProfessionalエディションが動作するユーザーデバイスでサポートされます。

サーバーでホストされるアプリケーションへのスマートカードによるパススルー認証は、Windows Server 2008およびWindows Server 2012が動作するサーバーでサポートされます。

サーバーでホストされるアプリケーションへのスマートカードパススルー認証を使用するには、サイトの認証方法としてスマートカードパススルーを構成するときにKerberosを有効にする必要があります。

注：スマートカードによるパススルー認証を使用できるかどうかは、さまざまな要因により決定されます。

- パススルー認証に関する組織のセキュリティポリシー。
- ミドルウェアの種類と構成。
- スマートカードリーダーの種類。
- ミドルウェアのPINキャッシュポリシー。

スマートカードによるパススルー認証は、Citrix StoreFront上で構成します。詳しくは、StoreFrontドキュメントの「[認証サービスの構成](#)」を参照してください。

## シングルサインオン

「シングルサインオン」とは、仮想デスクトップやアプリケーションの起動時にパススルー認証を実行する機能を指します。この機能を「ドメイン参加のStoreFront直接アクセス」および「ドメイン参加のNetScaler経由のStoreFrontアクセス」のスマートカード展開で使用して、ユーザーがPINを入力する回数を減らすことができます。これらの種類の展開でシングルサインオンを使用するには、StoreFrontサーバー上default.icaで以下のパラメーターを編集します。

- ドメイン参加のStoreFront直接アクセス — DisableCtrlAltDelをOffに設定します。
- ドメイン参加のNetScaler経由のStoreFrontアクセス — UseLocalUserAndPasswordをOnに設定します。

これらのパラメーター設定について詳しくは、StoreFrontまたはNetScaler Gatewayのドキュメントを参照してください。

シングルサインオン機能を使用できるかどうかは、以下を含むさまざまな要因により決定されます。

- シングルサインオンに関する組織のセキュリティポリシー。
- ミドルウェアの種類と構成。
- スマートカードリーダーの種類。
- ミドルウェアのPINキャッシュポリシー。

注：スマートカードリーダーが接続されたマシン上のVirtual Delivery Agent (VDA) にユーザーがログオンすると、前回使用された認証方法（スマートカードまたはパスワードなど）の画面が開く場合があります。この結果、シングルサインオンが効な場合はシングルサインオン用の画面が開きます。この画面ではログオンできないため、ユーザーは [ユーザーの切り替え] をクリックしてほかの画面を開く必要があります。

# SSL

Jan 04, 2017

XenAppまたはXenDesktopのサイトのSecure Sockets Layer (SSL) セキュリティプロトコルを構成するには、以下の手順が必要です。

- サーバー証明書を手入して、すべてのDelivery Controller上にインストールして登録します。さらに、SSL証明書のポート構成を行います。詳しくは、[SSLサーバー証明書のControllerへのインストール](#)を参照してください。

必要な場合は、ControllerでHTTPおよびHTTPSトラフィック用に使用されるポートを変更することもできます。

- ユーザーとVirtual Delivery Agent (VDA) 間のSSL接続を有効にします。これを行うには、以下のタスクが必要です。
  - VDAがインストールされたマシン上でSSLを構成します (便宜上、VDAがインストールされたマシンをここでは「VDA」と呼びます)。提供されているPowerShellスクリプトを使用したり、手作業で構成したりすることができます。一般的な情報については、「[VDA上のSSL設定について](#)」を参照してください。詳しくは、「[VDA上のSSL構成: PowerShellスクリプトの使用](#)」および「[VDA上のSSL構成: 手作業による構成](#)」を参照してください。

- VDAが追加されているデリバリーグループでSSLを構成します。これを行うには、StudioでいくつかのPowerShellコマンドレットを実行します。詳しくは、[デリバリーグループのSSLの構成](#)を参照してください。

以下の要件および考慮事項があります。

- ユーザーとVDA間のSSL接続を有効にするのは、XenApp 7.6サイト、XenDesktop 7.6サイト、およびこれ以降のリリースでのみ必要です。
- デリバリーグループおよびVDA上のSSLは、コンポーネントのインストール、サイトの作成、およびマシンカタログとデリバリーグループの作成を行った後で構成します。
- デリバリーグループでSSLを構成するには、Controllerのアクセス規則を変更するための権限が必要です。すべての管理権限を実行できる管理者には必要な権限が付与されています。
- VDA上のSSLを構成するには、そのマシン上のWindows管理者権限が必要です。
- VDAを以前のバージョンからアップグレードしてSSLを構成する場合は、アップグレード前にすべてのSSLリソースソフトウェアをアンインストールしておく必要があります。
- PowerShellスクリプトでは、静的に割り当てられるVDA上のSSLを構成できます。Machine Creation ServicesまたはProvisioning ServicesでプロビジョニングされてプールされるVDAは再起動時マシンイメージがリセットされるため、PowerShellスクリプトでSSLを構成することはできません。

Windowsレジストリの編集が必要なタスクを行う場合の注意事項。

注意: レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があります。Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

サイトデータベース接続のSSLを有効にする方法については、[CTX137556](#)を参照してください。

## SSLサーバー証明書のControllerへのインストール

HTTPS接続を使用する場合、XML Serviceはサーバー証明書を使用したSSL機能をサポートしますが、クライアント証明書はサポートしません。サーバー証明書を手入してController上にインストールおよび登録し、SSL証明書のポート構成を行うには、以下のタスクが必要です。

- ControllerにIISがインストールされている場合は、<https://technet.microsoft.com/ja-jp/library/cc771438%28v=ws.10%29.aspx>の手順に従います。
- ControllerにIISがインストールされていない場合は、以下の方法で証明書を構成します。

1. <http://blogs.technet.com/b/pki/archive/205/09/08/how-to-create-a-web-server-ssl-certificate-manually.aspx>を参照し、SSLサーバー証明書を取得してController上にインストールします。証明書要求ツール (Certreq) については、[http://technet.microsoft.com/ja-jp/library/cc736326\(WS.10\).aspx](http://technet.microsoft.com/ja-jp/library/cc736326(WS.10).aspx)を参照してください。PowerShellスクリプトを使用してVDA上のSSLを構成するときにSSL証明書の拇印を指定しない場合は、証明書ストアの [ローカルコンピューター] > [個人] > [証明書] に適切な証明書が配置されていることを確認してください。この場所に複数の証明書が配置されている場合、最初に検出されたものが使用されます。

2. 証明書でのポートの構成については、<http://msdn.microsoft.com/ja-jp/library/ms733791%28v=vs.110%29.aspx>を参照してください。

## HTTPまたはHTTPSポートの変更

デフォルトでは、XML ServiceはHTTPトラフィックにはポート80を、HTTPSトラフィックにはポート443を使用します。これらのポート番号を変更することもできますが、信頼されないネットワークにControllerを露出させる場合のセキュリティ上のリスクについて考慮してください。デフォルト構成を変更する場合は、スタンドアロンのStoreFrontサーバーを使用することをお勧めします。Controllerで使用されるデフォルトのHTTPまたはHTTPSポートを変更するには、StudioでBrokerService.exe -WIPOORT <http-port> -WISSLPORT <https-port>を実行します。

ここで、<http-port>はHTTPトラフィックのポート番号で、<https-port>はHTTPSトラフィックのポート番号です。

注: ポートが変更されると、ライセンスの互換性およびアップグレードに関するメッセージがStudioに表示されます。この問題を解決するには、以下のPowerShellコマンドレットを順に実行してサービスインスタンスを再登録してください。

```
Get-ConfigRegisteredServiceInstance -ServiceType Broker -Binding XML_HTTPS | Unregister-ConfigRegisteredServiceInstance Get-BrokerServiceInstance | where Binding -eq "XML_HTTPS" | Register
```

## HTTPSトラフィックのみに制限する

HTTPトラフィックがXML Serviceで無視されるように構成するには、Controller上のレジストリHKEY\_LOCAL\_MACHINE\Software\Citrix\DesktopServer\で以下の値を設定してからBroker Serviceを再起動します。

HTTPトラフィックを無視するには、XmlServicesEnableNonSslを0にします。

同様に、HTTPSトラフィックを無視するレジストリ値であるXmlServicesEnableSslも存在します。これは0に設定しないでください。

## VDA上のSSL設定について

VDA上のSSLを構成すると、インストールされているSSL証明書のアクセス権が変更され、その証明書の秘密キーに対する読み取り権限がICA Serviceに付与されます。ICA Serviceには、以下の情報が供されます。

- 証明書ストア内のどの証明書がSSLで使用されるのか。
- どのTCPポートがSSL接続で使用されるのか。  
Windowsファイアウォールを使用する環境では、このTCPでの着信接続が許可されている必要があります。PowerShellスクリプトを使用する場合は、このファイアウォール規則が自動的に構成されます。
- どのバージョンのSSLプロトコルが許可されるのか。  
サポートされるSSLプロトコルのバージョンは、低いものからSSL 3.0、TLS 1.0、TLS 1.1、およびTLS 1.2です。サポートされるSSLプロトコルを指定するときは、許可する最低バージョンを指定します。

たとえば、最低バージョンとしてTLS 1.1を指定すると、TLS 1.1およびTLS 1.2のプロトコルを使用した接続が許可されます。最低バージョンとしてSSL 3.0を指定すると、サポートされるSSLプロトコルのすべてのバージョンが許可されます。最低バージョンとしてTLS 1.2を指定すると、TLS 1.2の接続のみが許可されます。

- どのSSL暗号が許可されるのか。  
暗号スイート (Cipher Suite) とは、一般的なSSL暗号の一覧を指します。クライアントがVDAに接続するときは、そのクライアントがサポートするSSL暗号の一覧をVDAに送信します。VDA側で

は、構成済みの暗号スイート内にクライアントのいずれかのSSL暗号と一致するものがあるかどうかチェックされ、あった場合にのみ接続が確立されます。一致するSSL暗号がない場合、その接続はVDAにより拒否されます。

サポートされている暗号スイートはGOV (emment)、COM (mercial)、およびALLの3つです。これらの暗号スイートに含まれる暗号は、WindowsのFIPSモードによって異なります。WindowsのFIPSモードについては、<http://support.microsoft.com/kb/811833>を参照してください。次の表は、各暗号スイートに含まれる暗号を示しています。

| SSL暗号スイート | GOV | COM | すべて | GOV | COM | すべて |
|-----------|-----|-----|-----|-----|-----|-----|
| FIPSモード   | Off | Off | Off | On  | On  | On  |
| RSA_KEYX  | ○   | ○   | ○   | ○   | ○   | ○   |
| RSA_SIGN  | ○   | ○   | ○   | ○   | ○   | ○   |
| 3DES      | ○   |     | ○   | ○   |     | ○   |
| RC4       |     | ○   | ○   |     |     |     |
| MD5       | ○   | ○   | ○   |     |     |     |
| SHA       | ○   | ○   | ○   | ○   | ○   | ○   |
| SHA_256   | ○   | ○   | ○   | ○   | ○   | ○   |
| SHA_384   | ○   | ○   | ○   | ○   | ○   | ○   |
| SHA_512   | ○   | ○   | ○   | ○   | ○   | ○   |
| AES       | ○   | ○   | ○   | ○   | ○   | ○   |

SSLを構成したVDAと構成していないVDAを同一デリバリーグループ内で混在させることはできません。デリバリーグループのSSLを構成する前に、そのグループに属しているすべてのVDA上でSSL構成を完了しておく必要があります。

#### VDA上のSSL構成：PowerShellスクリプトの使用

VDA上でEnable-VdaSSL.ps1スクリプトを実行すると、そのVDAでのSSLリスナーを有効または無効にできます。このスクリプトは、インストールメディアのSupport > Tools > SslSupportフォルダーに収録されています。

スクリプトでSSLを有効にする場合、指定したSSL TCPポートについての既存のWindowsファイアウォール規則がすべて無効になります。その後で、ICA Serviceがそのポートで着信接続を受け入れるための新しい規則が追加されます。また、スクリプトにより以下のWindowsファイアウォール規則が無効になります。

- Citrix ICA (デフォルトで1494)
- Citrix CGP (デフォルトで2598)
- Citrix WebSocket (デフォルトで8008)

この結果、ユーザーはSSLでのみ接続できるようになり、Raw ICA、CGP、またはWebSocketで接続することはできなくなります。

このスクリプト内には、以下の構文および使用例が記載されています。Notepad++などのツールを使用してこれらを参照できます。

-Enableまたは-Disableパラメーターの指定は必須です。そのほかのパラメーターはオプションです。

#### 構文

```
Enable-VdaSSL [-Enable | -Disable] [-SSLPort <port>] [-SSLMinVersion "<min-ssl-version>"] [-SSLCipherSuite "<suite>"] [-CertificateThumbPrint "<thumbprint>"]
```

| パラメーター                                | 説明                                                                                                                                                              |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -Enable                               | SSLリスナーをVDA上にインストールして有効にします。このパラメーターまたは-Disableパラメーターのいずれかを指定する必要があります。                                                                                         |
| -Disable                              | VDA上のSSLリスナーを無効にします。このパラメーターまたは-Enableパラメーターのいずれかを指定する必要があります。このパラメーターを指定した場合、ほかのパラメーターは無視されます。                                                                 |
| -SSLPort <port>                       | SSLポート指定します。デフォルト：443                                                                                                                                           |
| -SSLMinVersion "<min-ssl-version>"    | 許可されるSSLプロトコルの最低バージョンを二重引用符で囲んで指定します。有効な値："SSL_3.0"、"TLS_1.0"、"TLS_1.1"、および"TLS_1.2"。デフォルト："TLS_1.0"                                                            |
| -SSLCipherSuite "<suite>"             | SSL暗号スイートを二重引用符で囲んで指定します。有効な値："GOV"、"COM"、および"ALL"。デフォルト："ALL"                                                                                                  |
| -CertificateThumbPrint "<thumbprint>" | 証明書ストア内のSSL証明書の拇印を二重引用符で囲んで指定します。通常、証明書ストア内に複数の証明書が配置されている場合にこのパラメーターを使用します。この拇印により、スクリプトで適切な証明書が選択されます。デフォルト：証明書ストアの [ローカルコンピューター] > [個人] > [証明書] で最初に検出された証明書 |

## 例

次のスクリプトでは、SSLリスナーをインストールして有効にします。すべてのオプションパラメーターのデフォルト値が適用されます。

```
Enable-VdaSSL -Enable
```

次のスクリプトでは、SSLリスナーをインストールして有効にして、SSLポートとして400、暗号スイートGOV、およびSSLプロトコルの最低バージョンとしてTLS 1.2を設定します。

```
Enable-VdaSSL -Enable -SSLPort 400 -SSLMinVersion "TLS_1.2" -SSLCipherSuite "GOV"
```

次のスクリプトでは、VDA上のSSLリスナーを無効にします。

```
Enable-VdaSSL -Disable
```

### VDA上のSSL構成：手作業による構成

VDA上のSSLを手作業で構成するには、SSL証明書の秘密キーに対する読み取り権限をVDA上のNT SERVICE\PorticaService (VDA for Windows Desktop OSの場合) またはNT SERVICE\TermService (VDA for Windows Server OSの場合) に付与します。VDAがインストールされたマシン上で、以下の手順を行います。

1. Microsoft管理コンソール (MMC) を起動します ( [スタート] > [ファイル名を指定して実行] mmc.exe ) 。
2. MMCに証明書スナップインを追加します。
  1. [ファイル] > [スナップインの追加と削除] の順に選択します。
  2. [証明書] を選択して [追加] をクリックします。
  3. [このスナップインで管理する証明書] で [コンピューターアカウント] をクリックし、 [次へ] をクリックします。
  4. [このスナップインで管理するコンピューター] で [ローカルコンピューター] をクリックし、 [完了] をクリックします。
3. コンソールツリーの [証明書 (ローカルコンピューター)] > [個人] > [証明書] で証明書を右クリックして、 [すべてのタスク] > [秘密キーの管理] の順に選択します。
4. [<FriendlyName> private keysのアクセス許可] ダイアログボックス (<FriendlyName>はSSL証明書の名前) が開きます。以下のいずれかのサービスを追加して、 [読み取り] アクセスを許可します。
  - VDA for Windows Desktop OSでは「PORTICASERVICE」
  - VDA for Windows Server OSでは「TERMSERVICE」
5. SSL証明書をダブルクリックします。 [証明書] ダイアログボックスの [詳細] タブをクリックして、一番下までスクロールします。 [拇印] をクリックします。
6. regeditを実行して、HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\Wds\icwdを開きます。
  1. SSL Thumbprintキーを編集して、SSL証明書の拇印の値をバイナリ値にコピーします。 [バイナリ値の編集] ダイアログボックスでは、不明な項目 (「0000」や特殊文字など) は無視して構いません。
  2. SSLEnabledキーを編集して、DWORD値を1に変更します (このDWORD値を0にするとSSLが無効になります) 。
  3. このレジストリパスでは、必要に応じて以下のデフォルト値を変更できます。
    - SSLPortのDWORD値 - SSLポート番号。デフォルト：443。
    - SSLMinVersionのDWORD値 - 1 = SSL 3.0、2 = TLS 1.0、3 = TLS 1.1、4 = TLS 1.2。デフォルト：2 (TLS 1.0) 。
    - SSLCipherSuiteのDWORD値 - 1 = GOV、2 = COM、3 = ALL。デフォルト：3 (ALL) 。
7. デフォルトの443以外のSSL TCPポートを使用する場合は、そのポートがWindowsファイアウォールで開放されていることを確認します (Windowsファイアウォールで受信規則を作成するときは、 [接続を許可する] および [有効] が選択されていることを確認してください) 。
8. ほかのアプリケーションやサービスなど (IISなど) がそのSSL TCPポートを使用していないことを確認します。
9. VDAs for Windows Server OSの場合は、変更を適用するためのマシンを再起動します。VDA for Windows Desktop OSのマシンを再起動する必要はありません。

### デリバリーグループのSSLの構成

SSL接続を構成したVDAを含んでいるすべてのデリバリーグループで、以下の手順を行います。

1. StudioからPowerShellコンソールを開きます。
2. asnp Citrix.\*を実行してCitrix製品のコマンドレットをロードします。
3. Get-BrokerAccessPolicyRule -DesktopGroupName '<delivery-group-name>' | Set-BrokerAccessPolicyRule -HdxSslEnabled \$trueを実行します。  
ここで、<delivery-group-name>はSSLを構成するデリバリーグループの名前です。
4. Set-BrokerSite -DnsResolutionEnabled \$trueを実行します。

### トラブルシューティング

接続エラーが発生した場合は、VDAのシステムイベントログを確認してください。

Receiver for WindowsでSSL関連の接続エラー (1030など) が発生した場合は、Desktop Viewerを無効にしてから再試行してください。接続エラーは解決されませんが、SSLの問題についての情報 (証明機関に証明書を要求したときに正しくないテンプレートを使用したなど) が表示される場合があります。

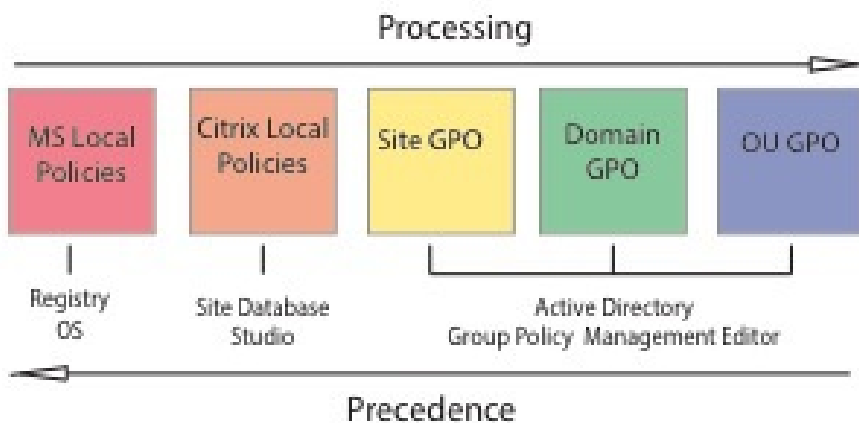


# ポリシー

Jan 04, 2017

ポリシーは構成可能な設定項目をグループ化したもので、特定のユーザー、デバイス、または接続の種類に対して特定のセッション、帯域幅、およびセキュリティ構成が適用されるように制御する目的で使用します。

これらのポリシーは、特定の物理マシン、仮想マシン、またはユーザーに割り当てることができます。ユーザーに適用する場合、ローカルレベルのアカウントを指定したりActive Directoryのセキュリティグループを指定したりできます。ポリシーに加えられる各設定項目では、特定の条件や規則を定義します。ポリシーを特定のオブジェクトに明示的に割り当てない場合、その設定はすべての接続に適用されます。



ポリシーは、ネットワークのさまざまなレベルに割り当てることができます。組織単位のGPOレベルに割り当てられたポリシーは、そのネットワークで最も優先されます。ドメインのGPOレベルのポリシーはサイトGPOレベルのポリシーよりも優先され、これらのポリシーはMicrosoftやCitrixのローカルポリシーよりも優先されます。

すべてのCitrixローカルポリシーはCitrix Studio上で作成および管理され、サイトデータベース内に格納されます。これに対し、グループポリシーはMicrosoftグループポリシー管理コンソール (GPMC) 上で作成および管理され、Active Directory内に格納されます。MicrosoftローカルポリシーはWindows上で作成され、レジストリ内に格納されます。

Studioのモデル作成ウィザードを使用すると、複数のテンプレートやポリシーの設定項目とその構成内容を比較してポリシーの競合や重複を避けることができます。また、GPMCを使用してGPOを設定して、ネットワークのさまざまなレベルのユーザーにそれらを適用できます。

これらのGPOはActive Directory内に保存され、セキュリティ上の理由から、通常IT担当者のみが設定を管理できます。

複数のポリシーの設定内容は、ポリシーの優先度や条件に基づいて統合されます。優先度のより高いポリシーの設定で [無効] または [禁止] が選択されている場合、優先度の低いポリシーで [有効] または [許可] が選択されていても、その設定内容は無視されます。未構成の設定項目は無視され、優先度の低いポリシーでの設定を上書きすることはありません。

ローカルポリシーとActive Directoryのグループポリシーの設定内容が競合する場合、優先されるポリシーは状況により異なります。

すべてのポリシーは、以下の順番で処理されます。

1. エンドユーザーがドメインの資格情報を使用してマシンにログオンする。
2. 資格情報がドメインコントローラーに送信される。
3. Active Directoryによりすべてのポリシー (エンドユーザー、エンドポイント、組織単位、およびドメイン) が適用され

る。

4. エンドユーザーがReceiverにログオンしてアプリケーションまたはデスクトップにアクセスする。
5. そのエンドユーザー、およびアプリケーションまたはデスクトップのホストマシンに適用されるCitrixポリシーとMicrosoftポリシーが処理される。
6. Active Directoryにより各ポリシー設定の優先度が決定され、エンドポイントデバイスのレジストリやリソースをホストしているマシンに適用されます。
7. エンドユーザーがアプリケーションまたはデスクトップからログオフする。そのエンドユーザー、およびアプリケーションまたはデスクトップのホストマシンに適用されるCitrixポリシーが非アクティブになる。
8. エンドユーザーがユーザーデバイスからログオフし、GPOユーザーポリシーが非アクティブになる。
9. エンドユーザーがユーザーデバイスをシャットダウンし、GPOマシンポリシーが非アクティブになる。

ユーザー、ユーザーデバイス、およびマシンのグループに割り当てるポリシーを作成する場合、そのグループの一部のメンバーで要件が異なるためにいくつかの設定項目で例外を設定しなければならない場合があります。この例外はStudioおよびGPMCでフィルターとして作成でき、これによりだれにどのポリシーが適用されるのかが決定されます。

## 関連トピック

- [ポリシーの使用](#)
- [ポリシーテンプレート](#)
- [ポリシーの作成](#)
- [ポリシーの比較、優先度、モデル作成、およびトラブルシューティング](#)
- [デフォルトのポリシー設定](#)
- [ポリシー設定リファレンス](#)

# ポリシーの使用

Jan 04, 2017

ユーザーのアクセスやセッション環境を制御するには、Citrixポリシーを構成します。Citrixポリシーを使用して、接続、セキュリティ、および帯域幅の設定を効率的に制御できます。ポリシーは、特定のグループのユーザー、デバイス、または接続の種類を対象に適用できます。1つのポリシーに複数の設定項目を選択して構成できます。

## Citrixポリシーを構成するツール

Citrixポリシーは、以下のツールを使用して構成します。

- **Studio** - グループポリシーの管理権限が付与されていないCitrix管理者は、Studioを使ってサイトのポリシーを作成します。Studioを使って作成されたポリシーはそのサイトのデータベースに保存され、仮想デスクトップをブローカーに登録するとき、またはユーザーが仮想デスクトップに接続するときにその仮想デスクトップに適用されます。
- **ローカルグループポリシーエディター (Microsoft管理コンソールのスナップイン)** - ネットワーク環境でActive Directoryが使用されており、グループポリシーの管理権限が付与されている場合は、グループポリシーエディターを使用してサイトのポリシーを作成できます。ここでの設定内容は、グループポリシー管理コンソールで指定するグループポリシーオブジェクト (GPO) に反映されます。

重要：VDAのControllerへの登録に関するものやMicrosoft App-Vサーバーに関するものなど、一部のポリシー設定を構成するには、グループポリシーエディターを使用する必要があります。

## ポリシーの処理順序と優先順位

グループポリシーの設定は、以下の順で処理されます。

1. ローカルのGPO
2. XenApp/XenDesktopサイトのGPO (サイトのデータベースに格納される)
3. サイトレベルのGPO
4. ドメインレベルのGPO
5. 組織単位

ただし、設定内容に競合が発生すると、最後に処理されるポリシーの設定により、先に処理されるポリシーの設定が上書きされることがあります。つまり、ポリシーの設定は以下の順番で優先されます。

1. 組織単位
2. ドメインレベルのGPO
3. サイトレベルのGPO
4. XenApp/XenDesktopサイトのGPO (サイトのデータベースに格納される)
5. ローカルのGPO

たとえば、営業部のユーザーがクライアント側のファイルをセッション内で使用できるようにするポリシー (Policy A) をCitrix管理者がStudioで作成し、同じユーザーに対してこの機能を禁止するポリシー (Policy B) をほかの管理者がグループポリシーエディターで作成したとします。この場合、営業部のユーザーが仮想デスクトップにログオンするとPolicy Bが適用され、Policy Aは無視されます。これは、ドメインレベルで処理されるPolicy Bが、XenApp/XenDesktopサイトのGPOレベルで処理されるPolicy Aよりも優先されるためです。

ただし、ユーザーがICAまたはリモートデスクトッププロトコル (RDP) セッションを開始する場合は、Active DirectoryやWindowsのリモートデスクトップセッションホストの構成ツールでの設定よりも、Citrixポリシーでの設定の方が優先されることに注意してください。これは、RDPクライアント接続で一般的に設定されている、デスクトップの壁紙、メニューのアニメーション化、ウィンドウの内容を表示したままドラッグする機能などにも当てはまります。

複数のポリシーを使用する場合は、競合する設定項目が正しく処理されるように優先順位を設定できます。詳しくは、[ポリ](#)

[シーの比較、優先度、モデル作成、およびトラブルシューティング](#)」を参照してください。

## Citrixポリシーの設定工程

ポリシーを設定する工程は次のとおりです。

1. ポリシーを作成します。
2. 設定項目を選択して構成します。
3. ポリシーをマシンやユーザーオブジェクトに割り当てます。
4. ポリシーの優先度を設定します。
5. Citrixグループポリシーモデル作成ウィザードを実行して、ポリシーの効果を確認します。

## Citrixポリシーと設定項目の使用

ローカルグループポリシーエディターでは、ポリシーと設定項目が [コンピューターの構成] ノードと [ユーザーの構成] ノードに表示されます。これらのそれぞれに [Citrix Policies] ノードがあります。このスナップインの使用方法については、Microsoft社のドキュメントを参照してください。

Studioでは、ポリシーやテンプレートの設定項目が機能に基づいて分類されています。たとえば、[Profile Management] カテゴリには、Profile managementのポリシー設定が含まれています。

- 「コンピューター設定」（マシンに適用される設定項目）は仮想デスクトップの動作を制御し、仮想デスクトップの起動時に適用されます。これらの設定項目は、仮想デスクトップにアクティブなユーザーセッションがない場合でも適用されます。「ユーザー設定」は、仮想デスクトップにICA接続する場合のユーザーエクスペリエンスを制御します。これらの設定項目は、ユーザーがICAを使って接続または再接続するたびに適用されます。ユーザーがRDPを使って接続したりコンソールに直接ログオンしたりする場合は適用されません。

ポリシー、設定項目、およびテンプレートを管理するには、Studioのナビゲーションペインで[ポリシー] を選択します。

- [ポリシー] タブには、すべての既存のポリシーが表示されます。ここでポリシーを選択すると、右側に [概要] タブ（名前、優先度、有効/無効の状態、および説明）、[設定] タブ（構成済みの設定項目の一覧）、および [割り当て先] タブ（ポリシーの適用対象のユーザーおよびマシンオブジェクト）が表示されます。詳しくは、「[ポリシーの作成](#)」を参照してください。
- [テンプレート] タブには、組み込みおよびカスタムのテンプレートが表示されます。ここでテンプレートを選択すると、右側に [説明] タブ（テンプレートの使用目的）、および [設定] タブ（構成済みの設定項目の一覧）が表示されます。詳しくは、「[ポリシーテンプレート](#)」を参照してください。
- [比較] タブでは、複数のポリシーやポリシーテンプレートの設定項目を比較することができます。環境に適した設定項目が構成されているかどうかを確認するときに、この機能を使用できます。詳しくは、「[ポリシーの比較、優先度、モデル作成、およびトラブルシューティング](#)」を参照してください。
- [モデル作成] タブでは、特定の接続シナリオでのCitrixポリシーの効果をシミュレートできます。詳しくは、「[ポリシーの比較、優先度、モデル作成、およびトラブルシューティング](#)」を参照してください。

ポリシーやテンプレートの設定項目を検索するには、以下の手順に従います。

1. ポリシーまたはテンプレートを選択します。
2. [操作] ペインの [ポリシーの編集] または [テンプレートの編集] を選択します。
3. [設定] ページで、設定項目の名前を入力します。

特定の製品バージョンや設定項目のカテゴリ（[帯域幅] など）を選択することで、検索範囲を限定できます。また、[選択項目のみを表示する] チェックボックスをオンにすると、そのポリシーで選択済みの設定項目のみが表示されます。すべての設定項目を検索対象にするには、[すべての設定] を選択します。

- ポリシーの設定項目を検索するには、以下の手順に従います。
  1. ポリシーを選択します。
  2. [設定] タブを選択し、設定項目の名前を入力します。

特定の製品バージョンや設定項目のカテゴリを選択することで、検索範囲を限定できます。すべての設定項目を検索対象にする

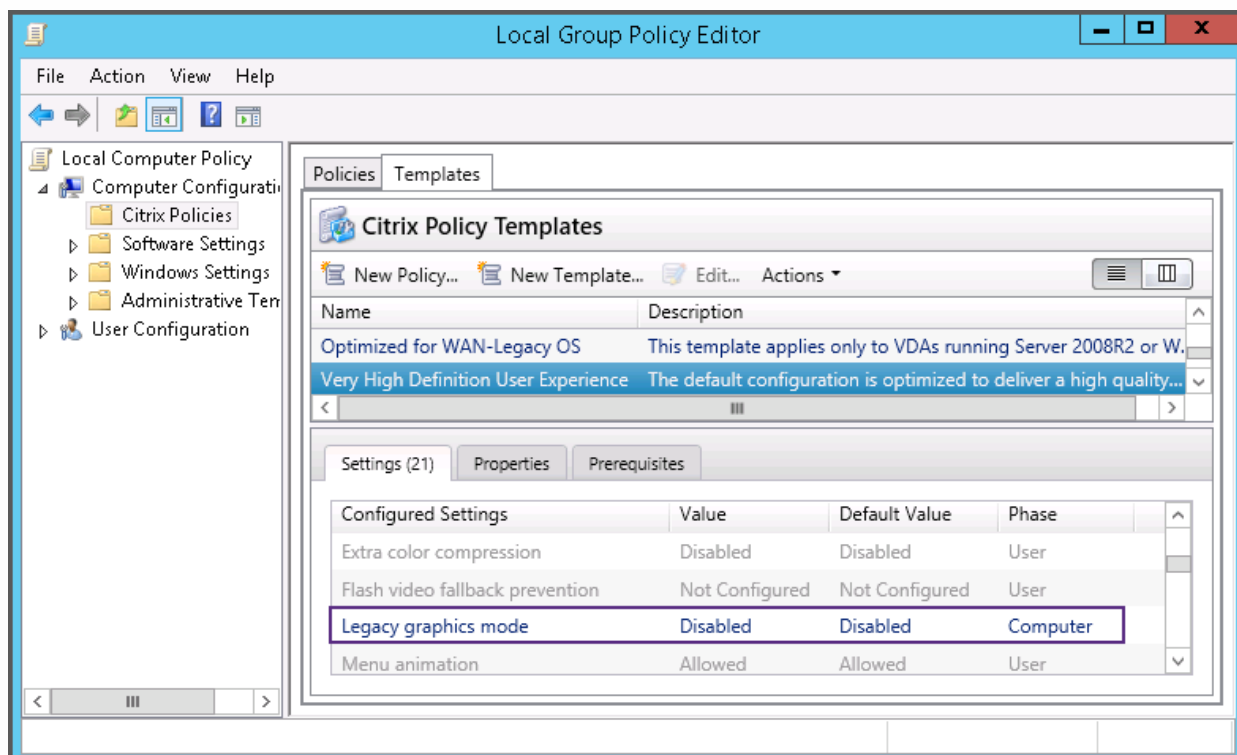
るには、 [すべての設定] を選択します。

いったんポリシーを作成したら、それは使用されるテンプレートとは全く無関係です。新しいポリシーの説明フィールドを使って、使用されるソーステンプレートを監視し続けることができます。

Studioでは、ユーザー、コンピューター、または両方の種類の設定のいずれかを含んでいるかどうかにかかわらず、ポリシーとテンプレートは単一の一覧に表示され、ユーザーとコンピューターの両方のフィルターを使って適用することができます。

グループポリシーエディターでは、コンピューターとユーザーの両方の種類の設定を含むテンプレートから作成された場合でも、コンピューターとユーザーは別々に適用される必要があります。この例では、 [コンピューターの構成] で [最高品位ユーザー エクスペリエンス] を使用することを選択しています。

- 従来のグラフィックモードは、このテンプレートから作成されるポリシーで使用されるコンピューター設定です。
- 灰色表示のユーザー設定は、このテンプレートから作成されるポリシーでは使用されません。



# ポリシーテンプレート

Jan 04, 2017

テンプレートは、事前定義された開始ポイントからポリシーを作成するためのソースです。組み込みCitrixテンプレートは、特定の環境またはネットワーク状況に対して最適化され、次のように使用できます。

- サイト間で共有する自分のポリシーおよびテンプレートを作成するためのソース。
- 結果を引用できるようになるため、展開環境間で結果をより簡単に比較するためのリファレンス。例: ""..when using Citrix template x or y.."
- テンプレートをインポートまたはエクスポートすることにより、Citrixサポートまたは信頼するサードパーティとポリシーを通信するための手段。

ポリシーテンプレートをインポートまたはエクスポートできます。追加のテンプレートと組み込みテンプレートの更新については、[CTX202000](#)を参照してください。

テンプレートを使ってポリシーを作成する場合の考慮事項については、[CTX202330](#)を参照してください。

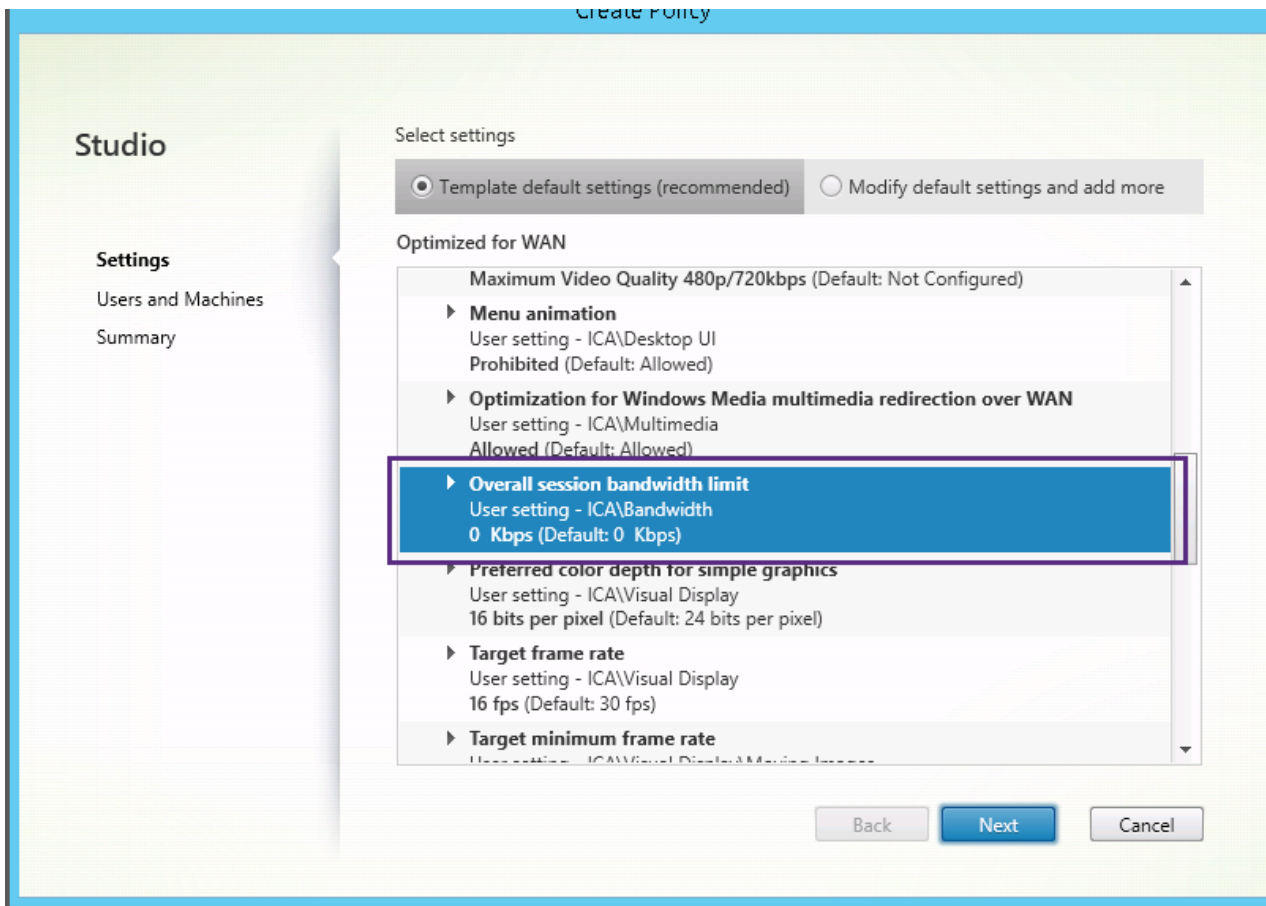
## 7.6 FP3の組み込みCitrixテンプレート

XenAppおよびXenDesktop 7.6 FP3 Group Policy Managementパッケージ (2.5.0.0) には、以前の組み込みCitrixテンプレートを置き換え、また強化する次のポリシーテンプレートが含まれています。

- **最高品位ユーザーエクスペリエンス**。このテンプレートは、デフォルトの設定を適用してユーザーエクスペリエンスを最適化します。このテンプレートは、複数のポリシーが優先順に処理されるシナリオで使用します。
- **高サーバースケーラビリティ**。サーバーリソースの浪費を避けるには、このテンプレートを適用します。このテンプレートはユーザーエクスペリエンスとサーバーのスケラビリティの均衡をとります。単一のサーバー上でホストできるユーザー数を増大させながら、良質なユーザーエクスペリエンスを提供します。このテンプレートは、グラフィックの圧縮にビデオコーデックを使用せず、サーバー側のマルチメディアレンダリングを防ぎます。
- **高サーバースケーラビリティ - レガシ OS**。この高サーバースケーラビリティテンプレートは、Server 2008 R2またはWindows 7以前が動作するVDAにのみ適用されます。このテンプレートは、これらのオペレーティングシステムでより効率的に機能する従来のグラフィックモードに依存します。
- **WANの最適化**。このテンプレートは、共有WAN接続を使用している支店や、低帯域幅接続を実行する遠隔地において、マルチメディアコンテンツがほとんどない視覚的に簡素なユーザーインターフェイスのアプリケーションにアクセスするタスクワーカーを対象としたものです。このテンプレートでは、ビデオ再生エクスペリエンスと一部のサーバースケーラビリティが帯域幅の効率性を最適化するため犠牲にされます。
- **WANの最適化 - レガシ OS**。このWANの最適化テンプレートは、Server 2008 R2 またはWindows 7以前が動作するVDAにのみ適用されます。このテンプレートは、これらのオペレーティングシステムでより効率的に機能する従来のグラフィックモードに依存します。
- **セキュリティと制御**。許容率が低い環境でのこのテンプレートの使用にはリスクがあります。XenAppおよびXenDesktopではデフォルトで有効な機能が最小化することになります。このテンプレートには、印刷、クリップボード、周辺デバイス、ドライブマッピング、ポートのリダイレクト、およびユーザーデバイス上のFlashアクセラレーションへのアクセスを無効にする設定があります。このテンプレートを適用すると、より多くの帯域幅が消費され、サーバーごとのユーザー密度が減ります。

組み込みCitrixテンプレートはそのデフォルトの設定のまま使用することをお勧めしますが、その設定には特定の推奨値はありません。たとえば、WANの最適化テンプレートにはセッション全体の最大帯域幅があります。この場合、テンプレートにより設定が公開され、これによって管理者はこの設定がそのシナリオに適用されようとしていることを理解します。

Create Policy



XenAppおよびXenDesktop 7.6FP3以前の展開環境（ポリシー管理およびVDA）を実行していて高サーバースケラビリティおよびWANの最適化テンプレートを必要とする場合、これらのテンプレートを適用するときはそのレガシOSバージョンを使用してください。

## 注意

組み込みテンプレートは、Citrixにより開発およびアップデートされます。これらのテンプレートを変更したり削除したりすることはできません。

### Studio使ったテンプレートの作成と管理

テンプレートをベースにした新しいテンプレートを作成するには：

1. Studioのナビゲーションペインで【ポリシー】を選択します。
2. 【テンプレート】タブを選択し、作成元のテンプレートを選択します。
3. 【操作】ペインの【テンプレートの作成】を選択します。
4. テンプレートのポリシー設定を選択して構成します。また、新しいテンプレートに不要な既存の設定は削除します。テンプレートの名前を入力します。

【完了】をクリックすると、新しいテンプレートが【テンプレート】タブに表示されます。

ポリシーをベースに新しいテンプレートを作成するには：

1. Studioのナビゲーションペインで [ポリシー] を選択します。
2. [ポリシー] タブを選択し、作成元のポリシーを選択します。
3. [操作] ペインの [テンプレートとして保存] を選択します。
4. テンプレートに含める新しいポリシー設定を追加して構成します。また、新しいテンプレートに不要な既存の設定は削除します。新しいテンプレートの名前と説明を入力し、[完了] をクリックします。

テンプレートをインポートするには：

1. Studioのナビゲーションペインで [ポリシー] を選択します。
2. [テンプレート] タブを選択し、[テンプレートのインポート] を選択します。
3. インポートするテンプレートを選択して、[開く] をクリックします。既存のものと同じ名前のテンプレートをインポートすると、既存のものを上書きするか、別名（自動的に生成されます）でインポートするかを選択できます。

テンプレートをエクスポートするには：

1. Studioのナビゲーションペインで [ポリシー] を選択します。
2. [テンプレート] タブを選択し、[テンプレートのエクスポート] を選択します。
3. テンプレートの保存先を指定して、[保存] をクリックします。

指定した場所にテンプレートファイル（拡張子.gpt）がエクスポートされます。

### グループポリシーエディターでテンプレートを作成および管理する

グループポリシーエディターの [コンピューターの構成] または [ユーザーの構成] を開きます。[ポリシー] ノードを開き、[Citrixポリシー] を選択します。以下の操作を行います。

| 目的                        | 手順                                             |
|---------------------------|------------------------------------------------|
| 既存のポリシーから新しいテンプレートを作成する   | [ポリシー] タブでポリシーを選択して [操作] > [テンプレートとして保存] を選択する |
| 既存のテンプレートから新しいポリシーを作成する   | [テンプレート] タブでテンプレートを選択して [新規ポリシー] をクリックする       |
| 既存のテンプレートから新しいテンプレートを作成する | [テンプレート] タブでテンプレートを選択して [新規テンプレート] をクリックする     |
| テンプレートをインポートする            | [テンプレート] タブで [操作] > [インポート] の順に選択する            |
| テンプレートをエクスポートする           | [テンプレート] タブで、[操作] > [エクスポート] の順に選択する           |
| テンプレートに構成済みの設定項目を確認する     | [テンプレート] タブでテンプレートを選択して [設定項目] タブをクリックする       |
| テンプレートのプロパティを確認する         | [テンプレート] タブでテンプレートを選択して [プロパティ] タブをクリックする      |
| テンプレートの必須条件を確認する          | [テンプレート] タブでテンプレートを選択して [必須条件] タブをクリックする       |

### 管理者の委任機能とテンプレート



ポリシーテンプレートは、ポリシー管理パッケージがインストールされたマシン上に格納されます。このマシンは、Delivery Controllerマシンかグループポリシーオブジェクト管理マシンのいずれかで、XenAppおよびXenDesktopサイトのデータベースではありません。これはつまり、ポリシーテンプレートファイルへのアクセスはWindowsの管理アクセス許可により制御され、サイトの委任された管理タスクの委任機能や管理スコープは考慮されません。

このため、たとえばサイトの読み取りのみの管理権限を持つ管理者がテンプレートを作成できます。ただし、テンプレートはローカルファイルであるため、ほかのマシン上のStudioには反映されません。

カスタムテンプレートは、それを作成するユーザーアカウントでのみ表示可能で、ユーザーのWindowsプロファイルに格納されます。これ以外のユーザーアカウントに対してもカスタムテンプレートを公開するには、そこからポリシーを作成するか、または共有の場所にエクスポートします。

# ポリシーの作成

Jan 04, 2017

ポリシーを作成する前に、そのポリシーの適用先となるユーザーまたはデバイスのグループを決定します。ユーザーの担当業務、接続の種類、ユーザーデバイス、または作業場所に応じてポリシーを適用できます。または、WindowsのActive Directoryのグループポリシーと同じ基準を使用できます。

グループに適用するポリシーを作成済みの場合は、別のポリシーを作成するのではなく、既存のポリシーの設定内容を編集することを検討してください。特定の設定内容を変更するため、または特定のユーザーを適用対象から除外するためだけに新しいポリシーを作成することは避けてください。

既存のポリシーテンプレートを基に新しいポリシーを作成し、必要に応じて設定項目をカスタマイズします。または、テンプレートを使用せずにポリシーを作成して、必要な設定項目を選択して構成します。

## ポリシー設定

ポリシーを設定するには、適用する設定項目を選択して値を構成します。デフォルトでは、ポリシーに追加されている設定項目はありません。設定項目を適用するには、ポリシーに追加する必要があります。

ポリシーのいくつかの設定項目では、次のオプションを指定します。

- [許可] または [禁止] を選択して、その設定項目により制御されるアクションを許可または禁止します。これらのアクションには、セッション内でのユーザーによる管理を許可したり禁止したりできるものがあります。たとえば、[メニューをアニメーション化する] 設定で [許可] を選択した場合、ユーザーがクライアント環境内でメニューのアニメーション化を制御できるようになります。
- [有効] または [無効] を選択して、その設定項目の機能を有効または無効にします。ここで無効にすると、より優先度の低いポリシーで [有効] を選択しても、その設定項目は有効になりません。

また、一部の設定項目は、それに依存する設定項目の効果を制御します。たとえば、[クライアントドライブリダイレクト] 設定により、クライアントデバイス側のドライブへのアクセスが制御されます。ユーザーがネットワークドライブにアクセスできるようにするには、この設定と [クライアント側ネットワークドライブ] 設定の両方で [許可] が選択されている必要があります。この場合、[クライアントドライブリダイレクト] 設定で [禁止] を選択すると、[クライアント側ネットワークドライブ] 設定で [許可] を選択しても、ユーザーがネットワークドライブにアクセスできなくなります。

通常、マシンの動作を制御する設定項目に対する変更内容は、仮想デスクトップが再起動したときまたはユーザーがログオンしたときに適用されます。また、ユーザーの機能を制御する設定項目は、そのユーザーの次回ログオン時に適用されます。Active Directory環境では、ポリシーが90分間隔で再評価され、仮想デスクトップが再起動したときまたはユーザーがログオンしたときに適用されます。

一部の設定項目では、ポリシーに追加するときに値を入力または選択します。[デフォルト値を使用する] チェックボックスをオンにすると、その設定項目のデフォルト値が適用され、ほかの値を設定できなくなります。[デフォルト値を使用する] チェックボックスをオンにする前に設定した値は無視されます。

ベストプラクティス：

- ポリシーの適用先として、個々のユーザーアカウントではなくグループアカウントを使用します。ポリシーの対象ユーザーを個々に追加したり削除したりするよりも、そのユーザーがグループアカウントに属しているかどうかで管理した方が効率的です。
- Windowsのリモートデスクトップセッションホストの構成ツールと重複または競合する設定項目を使用しないでください。リモートデスクトップセッションホストの構成ツールとCitrixポリシーで、同様の機能に対して異なる動作が設定されていると、予期せぬ問題が生じる場合があります。設定項目の有効/無効をできる限り統一しておくと、問題解決が容易に

なります。

- 使用しないポリシーは無効にしておきます。ポリシーに設定項目を追加しない場合でも、そのポリシーにより不要な処理が行われます。

## ポリシーの割り当て

ポリシーを特定のユーザーやマシンオブジェクトに割り当てると、設定した条件や規則に基づいてポリシーが接続に適用されます。通常、1つのポリシーに複数の割り当てを指定して、複数の条件を組み合わせることができます。割り当てを指定しない場合、そのポリシーはすべての接続に適用されます。

次の表は、使用可能な割り当ての一覧です。

| 割り当て名              | ポリシーの適用対象                                                                                                                                                                                                                                                          |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| アクセス制御             | セッションに接続するときのアクセス制御条件。 <ul style="list-style-type: none"><li>● 接続の種類 - 接続がNetScaler Gateway経由かどうかを指定します。</li><li>● NetScaler Gatewayファーム名 - NetScaler Gateway仮想サーバーの名前を指定します。</li><li>● Access Gateway Cluster - 使用するエンドポイント解析ポリシーまたはセッションポリシーの名前を入力します。</li></ul> |
| Citrix CloudBridge | ユーザーセッションがCitrix CloudBridge経由で起動されたかどうか。<br>注：ポリシーに追加できる [Citrix CloudBridge] 割り当ては1つのみです。                                                                                                                                                                        |
| クライアントのIPアドレス      | セッションに接続するクライアントデバイスのIPアドレス。 <ul style="list-style-type: none"><li>● IPv4の例：12.0.0.0, 12.0.0.*, 12.0.0.1-12.0.0.70, 12.0.0.1/24</li><li>● IPv6の例：2001:0db8:3c4d:0015:0:0:abcd:ef12, 2001:0db8:3c4d:0015::/54</li></ul>                                             |
| クライアント名            | ユーザーデバイスの名前。 <ul style="list-style-type: none"><li>● 完全一致：ClientABCName</li><li>● 使用するワイルドカード：Client*Name</li></ul>                                                                                                                                                |
| Delivery Group     | 所属するデリバリーグループ。                                                                                                                                                                                                                                                     |
| デリバリーグループの種類       | 実行されるデスクトップまたはアプリケーションの種類。プライベートデスクトップ、共有デスクトップ、プライベートアプリケーション、または共有アプリケーションから選択します。                                                                                                                                                                               |
| 組織単位 (OU)          | 組織単位。                                                                                                                                                                                                                                                              |
| タグ                 | マシンのタグ。<br><br>注：タグを使用する場合にポリシーを確実に正しく適用するには、 <a href="#">CTX142439</a> からHotfixを入手してインストールしてください。                                                                                                                                                                 |

| 割り当て名またはグループ | ポリシーの適用対象グループ名。 |
|--------------|-----------------|
|              |                 |

ユーザーがログオンするときに、その接続の条件に一致するすべてのポリシーが検出されます。検出されたポリシーは優先順に処理されます。このとき、ポリシー間で重複している設定がある場合は、最も優先度の高いポリシーの内容が適用されます。たとえば、優先度の高いポリシーの設定で「無効」が選択されている場合、優先度の低いポリシーの同じ設定で「有効」が選択されていても、その設定には「無効」が適用されます。構成されていないポリシー設定は無視されます。

**重要：**グループポリシー管理コンソールを使ってActive DirectoryポリシーとCitrixポリシーの両方を構成する場合、割り当ておよび設定が意図したとおりに適用されない場合があります。詳しくは、[CTX127461](#)を参照してください。

「Unfiltered」という名前のポリシーはデフォルトで提供されています。

- Studioを使用してCitrixポリシーを管理する場合は、Unfilteredポリシーに追加する設定がそのサイトのすべてのサーバー、仮想デスクトップ、および接続に適用されます。
- ローカルグループポリシーエディターを使用してCitrixポリシーを管理する場合は、そのポリシーのグループポリシーオブジェクト（GPO）スコープに属するすべてのサイトおよび接続にUnfilteredポリシーの設定が適用されます。たとえば、営業部署の組織単位に大阪支社のすべての営業メンバーを含んでいるSales-OSKというGPOがある場合に、いくつかのユーザーポリシー設定を追加したUnfilteredポリシーをSales-OSKに設定します。ここで大阪支社の営業部長がサイトにログオンすると、この部長はSales-OSK GPOのメンバーなので、Unfilteredポリシーのすべての設定がセッションに適用されます。

割り当ての「モード」によっても、そのポリシーの適用先が異なります。割り当てのモードとして「許可」（デフォルト）が設定されている場合、その割り当て条件にマッチした接続にのみポリシーが適用されます。割り当てのモードとして「禁止」が設定されている場合、その割り当て条件にマッチしない接続にのみポリシーが適用されます。以下の例では、複数の割り当てを追加したCitrixポリシーで、割り当てのモードがどのように適用されるかについて説明します。

- **例：同じ種類の割り当てでモードが異なる場合**- ポリシーに同じ種類の割り当てを追加し、一方を「許可」にしてもう一方を「禁止」にした場合、「禁止」を設定した割り当てが優先されます。例：  
Policy 1に以下の割り当てを追加します。

- Assignment Aは営業部署のグループアカウントに適用される割り当てで、「許可」を設定します。
- Assignment Bは営業部長のアカウントに適用される割り当てで、「禁止」を設定します。

ここで営業部長がログオンした場合、営業部長が営業部署のグループアカウントに属していても、Assignment Bが「禁止」モードなのでこのPolicy 1は適用されません。

- **例：異なる種類の割り当てでモードが同じ場合**- ポリシーに異なる種類の複数の割り当てを追加し、すべての割り当てに「許可」を設定した場合、すべての種類の割り当てに一致しないとポリシーは適用されません。例：  
Policy 2に以下の割り当てを追加します。

- Assignment Cは営業部署のグループアカウントに適用される「ユーザーまたはグループ」割り当てで、「許可」を設定します。
- Assignment Dは社内ネットワークのアドレス（10.8.169.\*）に適用される「クライアントのIPアドレス」割り当てで、「許可」を設定します。

ここで営業部長が社内のオフィスからログオンした場合、上記2つの割り当てに合致するので、このPolicy 2が適用されます。

Policy 3に以下の割り当てを追加します。

- Assignment Eは営業部署のグループアカウントに適用される「ユーザーまたはグループ」割り当てで、「許可」を設定します。
- Assignment Fは特定のNetScaler Gateway接続に適用される「アクセス制御」割り当てで、「許可」を設定します。

ここで営業部長が社内のオフィスからログオンした場合、Assignment Fに合致しないので、このPolicy 3は適用されません。

## Studioでテンプレートから新しいポリシーを作成する

1. Studioのナビゲーションペインで [ポリシー] を選択します。
2. [テンプレート] タブを選択し、テンプレートを選択します。
3. [操作] ペインの [テンプレートからのポリシーの作成] を選択します。
4. デフォルトでは、テンプレートで指定されているすべての設定項目が新しいポリシーに追加されます (テンプレートのデフォルトの設定項目) が有効)。設定項目を変更する場合は、[デフォルトの設定項目を変更および追加する] をクリックして、必要に応じて設定項目を追加または削除します。
5. ポリシーの割り当て先として、以下のいずれかを選択します。
  - [選択したユーザーおよびマシンオブジェクト] をクリックして、ポリシーを適用するユーザーおよびマシンオブジェクトを選択します。
  - [サイト内のすべてのオブジェクトに割り当てる] をクリックします。これにより、サイト内のすべてのユーザーやマシンオブジェクトにこのポリシーが適用されます。
6. 「新しいポリシーの名前を入力するか、デフォルトの名前を使用します。経理部」や「リモートユーザー」など、ポリシーの適用対象に基づいて名前を付けると便利です。また、必要に応じて説明を入力します。  
新しいポリシーはデフォルトで有効になりますが、無効にすることもできます。ポリシーを作成して有効にすると、新たにログオンするユーザーに直ちに適用されます。既存のセッションには適用されません。無効にしたポリシーは適用されません。作成済みのポリシーに優先度を設定したり、設定項目を追加したりするときは、そのポリシーを一時的に無効にすることを検討してください。

## Studioで新しいポリシーを作成する

1. Studioのナビゲーションペインで [ポリシー] を選択します。
2. [ポリシー] タブをクリックします。
3. [操作] ペインの [ポリシーの作成] を選択します。
4. 必要な設定項目を追加して構成します。
5. ポリシーの割り当て先として、以下のいずれかを選択します。
  - [選択したユーザーおよびマシンオブジェクト] をクリックして、ポリシーを適用するユーザーおよびマシンオブジェクトを選択します。
  - [サイト内のすべてのオブジェクトに割り当てる] をクリックします。これにより、サイト内のすべてのユーザーやマシンオブジェクトにこのポリシーが適用されます。
6. 「新しいポリシーの名前を入力するか、デフォルトの名前を使用します。経理部」や「リモートユーザー」など、ポリシーの適用対象に基づいて名前を付けると便利です。また、必要に応じて説明を入力します。  
新しいポリシーはデフォルトで有効になりますが、無効にすることもできます。ポリシーを作成して有効にすると、新たにログオンするユーザーに直ちに適用されます。既存のセッションには適用されません。無効にしたポリシーは適用されません。作成済みのポリシーに優先度を設定したり、設定項目を追加したりするときは、そのポリシーを一時的に無効にすることを検討してください。

## グループポリシーエディターでポリシーを作成および管理する

グループポリシーエディターの [コンピューターの構成] または [ユーザーの構成] を開きます。[ポリシー] ノードを開き、[Citrixポリシー] を選択します。以下の操作を行います。

| 目的           | 手順                               |
|--------------|----------------------------------|
| 新しいポリシーの作成   | [ポリシー] タブの [新規] をクリックする          |
| 既存のポリシーを編集する | [ポリシー] タブでポリシーを選択して [編集] をクリックする |

|                         |                                                     |
|-------------------------|-----------------------------------------------------|
| 既存のポリシーの優先度を変更する<br>目的  | 「ポリシー」タブでポリシーを選択して [上げる] または [下げる] をクリックする<br>手順    |
| ポリシーの要約情報を表示する          | [ポリシー] タブでポリシーを選択して [情報] タブをクリックする                  |
| ポリシーの設定項目を表示して変更する      | [ポリシー] タブでポリシーを選択して [設定] タブをクリックする                  |
| ポリシーの割り当て先を表示して変更する     | [ポリシー] タブでポリシーを選択して [フィルター] タブをクリックする               |
| ポリシーを有効または無効にする         | [ポリシー] タブでポリシーを選択して [操作]、[有効] または [操作]、[無効] の順に選択する |
| 既存のテンプレートから新しいポリシーを作成する | [テンプレート] タブでテンプレートを選択して [新規ポリシー] をクリックする            |

# ポリシーの比較、優先度、モデル作成、およびトラブルシューティング

Jan 04, 2017

ユーザーの担当業務、作業場所、または接続の種類などのユーザーのニーズに応じて、複数のポリシーを作成できます。たとえば、セキュリティ上の理由から、機密性の高いデータを日常的に取り扱うユーザーグループのアクセスに、一定の制限を適用したい場合があります。この場合、ユーザーがローカルのクライアントドライブ上にファイルを保存することを禁止するポリシーを作成できます。また、そのユーザーグループの中にローカルドライブへのアクセスが必要なユーザーがいる場合は、そのユーザー専用のポリシーを作成してほかのポリシーよりも高い優先度を設定します。同じユーザーに複数のポリシーが適用される場合は、それらのポリシーに優先度を設定して、適用される設定内容を制御できます。

複数のポリシーを使用するときは、どのように優先度を設定するか、どのように特定のユーザーを対象から除外するか、およびポリシーが競合した場合にどの設定内容が最終的に適用されるかについて確認する必要があります。

通常、Citrixポリシーの設定は、サイト全体、またはDelivery Controllerやユーザーデバイス側で構成されている同様の設定よりも優先されます。ただし、暗号化レベルとシャドウ機能の設定については、オペレーティングシステムでの設定を含み、最も高い制限が適用されます。

Citrixポリシーは、オペレーティングシステム側で設定されているほかのポリシーとも関連して機能します。Citrix環境では、Active DirectoryやWindowsのリモートデスクトップセッションホストの構成ツールでの設定よりも、Citrixポリシーでの設定の方が優先されます。これは、RDP (Remote Desktop Protocol) クライアント接続で一般的に設定されている、デスクトップの壁紙、メニューのアニメーション化、ウィンドウの内容を表示したままドラッグする機能などにも当てはまります。また、[SecureICAの最低暗号化レベル] など、オペレーティングシステム側の設定と合致していなければならないものもあります。Citrixポリシー以外の機能でより高い暗号化レベルが設定されている場合、[SecureICAの最低暗号化レベル] 設定やアプリケーションやデスクトップごとに指定されている配信設定は無視されます。

たとえば、デリバリーグループを作成するときに指定する暗号化レベルは、その環境全体に対して設定されているレベルと同じである必要があります。

注：いわゆるダブルホップ環境における2つ目のホップにおいて、デスクトップOSのVDAがサーバーOSのVDAに接続すると、デスクトップOSのVDAでCitrixポリシーがユーザーデバイスのように機能します。たとえば、ユーザーデバイス上のイメージをキャッシュするようポリシーが設定されると、ダブルホップ環境における2つ目のホップに対してキャッシュされたイメージはデスクトップOSのVDAマシンでキャッシュされます。

## ポリシーおよびテンプレートの比較

Studioでは、複数のポリシーやポリシーテンプレートの設定項目を比較することができます。たとえば、環境に適した設定項目が構成されているかどうかを確認するときに、この機能を使用できます。また、そのポリシーやテンプレートの各設定項目の設定値を、デフォルトの値と比較することもできます。

1. Studioのナビゲーションペインで [ポリシー] を選択します。
2. [比較] タブをクリックし、[選択] をクリックします。
3. 比較するポリシーまたはテンプレートのチェックボックスをオンにします。[設定項目のデフォルト値と比較する] チェックボックスをオンにすると、各設定項目のデフォルト値が比較結果に追加されます。
4. [比較] をクリックすると、構成された設定項目とその設定値が一覧表示されます。
5. すべての設定項目を表示するには、[すべての設定項目を表示] を選択します。元の表示に戻るには、[共通の設定項目を表示] を選択します。

## ポリシーの優先度

複数のポリシーで設定内容が競合することを防ぐために、ポリシーに優先度を設定できます。ユーザーがログオンするときに、その接続の条件に一致するすべてのポリシーが検出されます。検出されたポリシーは優先度順に処理されます。このとき、ポリシー間で重複している設定項目がある場合は、最も優先度の高いポリシーの内容が適用されます。

Studioでは、ポリシーの優先度が数値で示されます。デフォルトでは、新しいポリシーに最低の優先度が設定されます。複数のポリシーで設定内容に矛盾が生じた場合は、優先度の高いポリシー（最高の優先度は「1」です）の設定が適用されます。同じ条件の接続に対して複数のポリシーが合致する場合は、各ポリシーに追加されている設定項目がポリシーの優先度、および各設定内容により統合され、「最終的に適用されるポリシー」が決定されます。優先度のより高いポリシーの設定で[無効]が選択されている場合、優先度の低いポリシーで[有効]が選択されていても、その設定内容は無視されます。ただし、[設定しない]が選択された設定項目は無視されるため、優先度の高いポリシーで[設定しない]が設定されている場合、その設定内容は無視され、優先度の低いポリシーの内容が適用されます。

1. Studioのナビゲーションペインで[ポリシー]を選択します。[ポリシー]タブが選択されていることを確認します。
2. ポリシーを選択します。
3. [操作]ペインの[優先度を低く]または[優先度を高く]を選択します。

## 例外

ユーザー、ユーザーデバイス、またはマシンに対して作成したポリシーの中に、そのグループの特定のユーザーに適用しない設定内容が含まれている場合は、以下の方法で例外を設定します。

- 例外処理が必要なグループメンバー用に新しいポリシーを作成して、ほかのポリシーより高い優先度を設定します。
- ポリシーに追加する割り当てのモードとして[禁止]を選択します。

割り当てのモードとして[禁止]を選択すると、その条件にマッチしない接続にのみポリシーが適用されます。以下の割り当てを指定したポリシーを例にして説明します。

- [クライアントのIPアドレス]割り当てで「208.77.88.\*」を指定して[許可]モードを選択
- [ユーザーまたはグループ]割り当てで特定のユーザーアカウントを指定して[禁止]モードを選択

この2つの割り当てが追加されたポリシーは、[クライアントのIPアドレス]割り当てで指定したアドレスのすべてのユーザーに適用されます。ただし、[ユーザーまたはグループ]割り当てで指定したユーザーには、IPアドレスが指定した範囲内であってもポリシーは適用されません。

## 接続に適用されるポリシーの確認

複数のポリシーが適用されるために、意図した設定が接続に反映されないことがあります。作成したポリシーよりも優先度の高いポリシーがあると、意図した設定内容が上書きされてしまいます。管理者は、ポリシーの優先度や追加されている設定項目を基に、最終的に適用される設定項目を確認することができます。

最終的に適用される設定項目を確認するには、以下の方法を使用します。

- Citrixグループポリシーモデル作成ウィザードを使用して、接続シナリオをシミュレートしてCitrixポリシーがどのように適用されるかを確認する。接続シナリオ条件（ドメインコントローラー、ユーザー、Citrixポリシーの割り当て、低速ネットワーク接続などの環境設定）を指定します。すると、その条件に基づいて、そのシナリオに適用されるCitrixポリシーの内容についてのレポートが生成されます。ドメインユーザーとしてControllerにログオンしている場合は、サイトポリシー設定とActive Directoryグループポリシーオブジェクト（GPO）の両方を使ってポリシーの結果セットが算出されます。
- グループポリシーの結果ウィザードで、特定のユーザーやControllerに適用されるCitrixポリシーのレポートを作成する。グループポリシーの結果ウィザードでは、現在の環境のGPOの状態を評価して、特定のユーザーやControllerにこれらのオブジェクト（Citrixポリシーを含む）がどのように適用されるかについてのレポートが生成されます。

Citrixグループポリシーモデル作成ウィザードは、Studioの[操作]ペインから起動できます。これらのツールは、Windowsのグループポリシー管理コンソールから起動できます。

グループポリシー管理コンソールからCitrixポリシーモデル作成ウィザードまたはグループポリシーの結果ウィザードを実行



する場合は、Studioで作成したサイトポリシー設定はポリシーの結果セットに含まれません。

ポリシーの管理にグループポリシー管理コンソールのみを使用している場合を除き、最も包括的なポリシーの結果セットを得るには、StudioからCitrixグループポリシーモデル作成ウィザードを起動することをお勧めします。

## Citrixグループポリシーモデル作成ウィザードの使用

Citrixグループポリシーモデル作成ウィザードを開くには、以下のいずれかを行います。

- Studioのナビゲーションペインで[ポリシー]を選択し、[モデル作成]タブを選択して[操作]ペインの[モデル作成ウィザードの起動]を選択します。
- グループポリシー管理コンソール (gpmc.msc) を起動して、コンソールツリーの[Citrixグループポリシーモデル作成]ノードを右クリックして[Citrixグループポリシーモデル作成ウィザード]を選択します。

ウィザードの指示に従って、シミュレーションで使用するドメインコントローラー、ユーザー、コンピューター、環境設定、およびCitrixフィルター条件を選択します。[完了]をクリックすると、モデル作成の結果のレポートが作成されます。Studioでは、中央ペインの[モデル作成]タブにレポートが表示されます。

レポートを表示するには、[モデル作成レポートの表示]を選択します。

## ポリシーのトラブルシューティング

複数のポリシーで、適用先として同じ割り当て（ユーザーアカウントやクライアントのIPアドレスなど）を指定することも可能です。この場合、これらのポリシーでの設定が競合すると、ポリシーが意図したとおりに適用されません。最終的に適用されるポリシーを確認するためにCitrixグループポリシーモデル作成ウィザードやグループポリシーの結果ウィザードを使用する場合、ユーザー接続にいずれのポリシーも適用されないことが判明することがあります。この場合、そのポリシーの割り当て条件に合致するユーザー接続が発生しても、いずれの設定項目も適用されません。以下の状況では、いずれのポリシーも適用されません。

- 割り当て条件に合致するポリシーがない場合。
- 割り当て条件に合致したポリシーに設定項目が追加されていない場合。
- 割り当て条件に合致したポリシーが無効になっている場合。

指定した条件の接続にポリシーが適用されるようにするには、以下の内容を確認します。

- そのポリシーが有効になっている。
- そのポリシーに追加した設定項目の内容が適切である。

# デフォルトのポリシー設定

Jan 04, 2017

次の表は、ポリシーの各設定項目のデフォルト設定と、適用されるVirtual Delivery Agent (VDA) のバージョンの一覧です。

## ICA

| 名前                        | デフォルト設定   | VDA                                                         |
|---------------------------|-----------|-------------------------------------------------------------|
| クライアントクリップボードリダイレクト       | 許可        | すべてのバージョンのVDA                                               |
| デスクトップの起動                 | 禁止        | VDA for Windows Server OS 7以降                               |
| ICAリスナー接続タイムアウト           | 120000ミリ秒 | VDA 5、5.5、5.6 Feature Pack 1、VDA for Windows Desktop OS 7以降 |
| ICAリスナーポートの番号             | 1494      | すべてのバージョンのVDA                                               |
| クライアント接続での非公開アプリケーションの起動  | 禁止        | VDA for Windows Server OS 7以降                               |
| クライアントクリップボードに書き込みを許可する形式 | 形式の指定なし   | VDA 7.6                                                     |
| クライアントクリップボードの書き込み制限      | 禁止        | VDA 7.6                                                     |
| セッションクリップボードの書き込み制限       | 禁止        | VDA 7.6                                                     |
| セッションクリップボードに書き込みを許可する形式  | 形式の指定なし   | VDA 7.6                                                     |

## ICA/Adobe Flashデリバリー/Flashリダイレクト

| 名前                        | デフォルト設定 | VDA         |
|---------------------------|---------|-------------|
| Flashビデオフォールバック防止         | 未構成     | VDA 7.6 FP3 |
| Flashビデオフォールバック防止エラー*.swf |         | VDA 7.6 FP3 |

## ICA/オーディオ

| 名前                | デフォルト設定      | VDA                           |
|-------------------|--------------|-------------------------------|
| オーディオプラグアンドプレイ    | 許可           | VDA for Windows Server OS 7以降 |
| 音質                | 高 - 高品位オーディオ | すべてのバージョンのVDA                 |
| クライアントオーディオリダイレクト | 許可           | すべてのバージョンのVDA                 |
| クライアントマイクリダイレクト   | 許可           | すべてのバージョンのVDA                 |

#### ICA/クライアントの自動再接続

| 名前               | デフォルト設定            | VDA |
|------------------|--------------------|-----|
| クライアントの自動再接続     | 許可                 | VDA |
| クライアントの自動再接続時の認証 | 認証を必要としない          | VDA |
| クライアントの自動再接続のログ  | 自動再接続イベントをログに記録しない | VDA |

#### ICA/帯域幅

| 名前                         | デフォルト設定 | VDA                                                                                        |
|----------------------------|---------|--------------------------------------------------------------------------------------------|
| オーディオリダイレクトの最大帯域幅 (Kbps)   | 0Kbps   | VDA                                                                                        |
| オーディオリダイレクトの最大帯域幅 (%)      | 0       | VDA                                                                                        |
| USBデバイスリダイレクトの最大帯域幅 (Kbps) | 0Kbps   | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| USBデバイスリダイレクトの最大帯域幅 (%)    | 0       | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| クリップボードリダイレクトの最大帯域幅 (Kbps) | 0Kbps   | すべてのバージョンのVDA                                                                              |
| クリップボードリダイレクトの最大帯域幅 (%)    | 0       | すべてのバージョンのVDA                                                                              |
| COMポートリダイレクトの最大帯域幅         | 0Kbps   | すべてのバージョンのVDA。VDA 7.xではレジストリで設定し                                                           |

|                                              |         |                                                                                            |
|----------------------------------------------|---------|--------------------------------------------------------------------------------------------|
| (Kbps)<br>名前<br>COMポートリダイレクトの最大帯域幅 (%)       | デフォルト設定 | ます<br>VDA<br>すべてのバージョンのVDA。VDA 7.xではレジストリで設定します                                            |
| ファイルリダイレクトの最大帯域幅 (Kbps)                      | 0Kbps   | すべてのバージョンのVDA                                                                              |
| ファイルリダイレクトの最大帯域幅 (%)                         | 0       | すべてのバージョンのVDA                                                                              |
| HDX MediaStreamマルチメディアアクセラレーションの最大帯域幅 (Kbps) | 0Kbps   | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| HDX MediaStreamマルチメディアアクセラレーションの最大帯域幅 (%)    | 0       | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| LPTポートリダイレクトの最大帯域幅 (Kbps)                    | 0Kbps   | すべてのバージョンのVDA。VDA 7.xではレジストリで設定します                                                         |
| LPTポートリダイレクトの最大帯域幅 (%)                       | 0       | すべてのバージョンのVDA。VDA 7.xではレジストリで設定します                                                         |
| セッション全体の最大帯域幅                                | 0Kbps   | すべてのバージョンのVDA                                                                              |
| プリンターリダイレクトの最大帯域幅 (Kbps)                     | 0Kbps   | すべてのバージョンのVDA                                                                              |
| プリンターリダイレクトの最大帯域幅 (%)                        | 0       | すべてのバージョンのVDA                                                                              |
| TWAINデバイスリダイレクトの最大帯域幅 (Kbps)                 | 0Kbps   | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| TWAINデバイスリダイレクトの最大帯域幅 (%)                    | 0       | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |

#### ICA/クライアントセンサー

| 名前                            | デフォルト設定 | VDA                                                                                    |
|-------------------------------|---------|----------------------------------------------------------------------------------------|
| クライアントデバイスの位置情報をアプリケーションで使用する | 禁止      | VDA 5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |

#### ICA/デスクトップUI

| 名前 | デフォルト設定 | VDA |
|----|---------|-----|
|    |         |     |

|                           |                           |                                           |
|---------------------------|---------------------------|-------------------------------------------|
| デスクトップコンポジションリダイレクト<br>名前 | 無効 (7.6 FP3以降)<br>デフォルト設定 | VDA 5.6、VDA for Desktop OS 7以降、VDA<br>VDA |
|                           | 有効 (5.6~7.6 FP2)          |                                           |
| デスクトップコンポジションの画質          | 中                         | VDA 5.6、VDA for Desktop OS 7以降、VDA        |
| デスクトップの壁紙                 | 許可                        | すべてのバージョンのVDA                             |
| メニューをアニメーション化する           | 許可                        | すべてのバージョンのVDA                             |
| ドラッグ中にウィンドウの内容を表示する       | 許可                        | すべてのバージョンのVDA                             |

#### ICA/エンドユーザーモニタリング

| 名前             | デフォルト設定 | VDA           |
|----------------|---------|---------------|
| ICA往復測定        | 有効      | すべてのバージョンのVDA |
| ICA往復測定間隔      | 15秒     | すべてのバージョンのVDA |
| アイドル接続のICA往復測定 | 無効      | すべてのバージョンのVDA |

#### ICA/デスクトップエクスペリエンス拡張

| 名前               | デフォルト設定 | VDA                           |
|------------------|---------|-------------------------------|
| デスクトップエクスペリエンス拡張 | 許可      | VDA for Windows Server OS 7以降 |

#### ICA/ファイルリダイレクト

| 名前                | デフォルト設定 | VDA           |
|-------------------|---------|---------------|
| クライアントドライブに自動接続する | 許可      | すべてのバージョンのVDA |
| クライアントドライブリダイレクト  | 許可      | すべてのバージョンのVDA |
| クライアント側固定ドライブ     | 許可      | すべてのバージョンのVDA |
| クライアント側フロッピードライブ  | 許可      | すべてのバージョンのVDA |

|                         |              |                                                                                            |
|-------------------------|--------------|--------------------------------------------------------------------------------------------|
| 名前<br>クライアント側ネットワークドライブ | 許可<br>フォルト設定 | VDA 5.5でのバージョンのVDA                                                                         |
| クライアント側光学式ドライブ          | 許可           | すべてのバージョンのVDA                                                                              |
| クライアント側リムーバブルドライブ       | 許可           | すべてのバージョンのVDA                                                                              |
| ホストからクライアントへのリダイレクト     | 無効           | VDA for Windows Server OS 7以降                                                              |
| クライアント側のドライブ文字を保持する     | 無効           | VDA 5、5.5、5.6 Feature Pack 1、VDA for Windows Desktop OS 7以降                                |
| クライアント側ドライブへの読み取り専用アクセス | 無効           | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| ユーザーフォルダーのリダイレクト        | 許可           | Web Interface環境でのみ。VDA for Windows Server OS 7以降                                           |
| 非同期書き込みを使用する            | 無効           | すべてのバージョンのVDA                                                                              |

#### ICA/グラフィック

| 名前                        | デフォルト設定        | VDA                                                                                        |
|---------------------------|----------------|--------------------------------------------------------------------------------------------|
| 表示メモリの制限                  | 65536KB        | VDA 5、5.5、5.6 Feature Pack 1、VDA for Windows Desktop OS 7以降                                |
| メモリが不足したときの表示モード          | 色数を下げる         | すべてのバージョンのVDA                                                                              |
| 動的ウィンドウプレビュー              | 有効             | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| イメージキャッシュ                 | 有効             | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| 従来のグラフィックモード              | 無効             | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降                            |
| 許可される最大表示色数               | 32ビット/<br>ピクセル | すべてのバージョンのVDA                                                                              |
| メモリ不足による表示品質の低下をユーザーに通知する | 無効             | VDA for Windows Server OS 7以降                                                              |
| キューイメージの破棄                | 有効             | すべてのバージョンのVDA                                                                              |

| 名前               | デフォルト設定 | VDA |
|------------------|---------|-----|
| ICA/グラフィック/キャッシュ |         |     |

| 名前          | デフォルト設定    | VDA                           |
|-------------|------------|-------------------------------|
| 固定キャッシュしきい値 | 3000000bps | VDA for Windows Server OS 7以降 |

#### ICA/Keep-Alive

| 名前                   | デフォルト設定                   | VDA           |
|----------------------|---------------------------|---------------|
| ICA Keep-Aliveタイムアウト | 60秒                       | すべてのバージョンのVDA |
| ICA Keep-Alive       | ICA Keep-Aliveメッセージを送信しない | すべてのバージョンのVDA |

#### ICA/ローカルアプリケーションアクセス

| 名前                     | デフォルト設定  | VDA                                                             |
|------------------------|----------|-----------------------------------------------------------------|
| ローカルアプリケーションのアクセスを許可する | 禁止       | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| URLリダイレクトのブラックリスト      | サイトの指定なし | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| URLリダイレクトのホワイトリスト      | サイトの指定なし | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |

#### ICA/モバイルデバイスでの動作

| 名前                      | デフォルト設定 | VDA                                                                                                                                       |
|-------------------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------|
| キーボードの自動表示              | 禁止      | VDA 5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降                                                    |
| タッチパネルでの操作に最適化されたデスクトップ | 許可      | VDA 5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降<br><br>この設定はデフォルトで無効になっており、Windows 10マシンでは使用できません。 |
| コンボボックスをデバイス側で表示する      | 禁止      | VDA 5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降                                                    |

## ICA/マルチメディア

| 名前                                        | デフォルト設定 | VDA                                                             |
|-------------------------------------------|---------|-----------------------------------------------------------------|
| ビデオ品質の制限                                  | 未構成     | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| マルチメディア会議                                 | 許可      | すべてのバージョンのVDA                                                   |
| WAN接続でのWindows Mediaマルチメディアリダイレクトの最適化     | 許可      | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| WAN接続でのWindows MediaマルチメディアリダイレクトでのGPUの使用 | 禁止      | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| ビデオ負荷管理ポリシー設定                             | 未構成     | VDA 7.6 FP3                                                     |
| Windows Mediaのクライアント側でのコンテンツ取得            | 許可      | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| Windowsメディアリダイレクト                         | 許可      | すべてのバージョンのVDA                                                   |
| Windows Mediaリダイレクトのバッファサイズ               | 5秒      | VDA 5、5.5、および5.6 Feature Pack 1                                 |
| Windows Mediaリダイレクトのバッファサイズ使用             | 無効      | VDA 5、5.5、および5.6 Feature Pack 1                                 |

## ICA/マルチストリーム接続

| 名前             | デフォルト設定     | VDA                                                                                        |
|----------------|-------------|--------------------------------------------------------------------------------------------|
| UDPを使用したオーディオ  | 許可          | VDA for Windows Server OS 7以降                                                              |
| オーディオUDPポートの範囲 | 16500、16509 | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |



|                   |                                           |                                                                                            |
|-------------------|-------------------------------------------|--------------------------------------------------------------------------------------------|
| マルチポートポリシー        | プライマリポート<br>デフォルト設定<br>(2598) に優先度<br>[高] | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| マルチストリームコンピューター設定 | 無効                                        | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| マルチストリームユーザー設定    | 無効                                        | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |

#### ICA/ポートリダイレクト

| 名前                  | デフォルト設定 | VDA                                |
|---------------------|---------|------------------------------------|
| クライアントCOMポートを自動接続する | 無効      | すべてのバージョンのVDA。VDA 7.xではレジストリで設定します |
| クライアントLPTポートを自動接続する | 無効      | すべてのバージョンのVDA。VDA 7.xではレジストリで設定します |
| クライアントCOMポートリダイレクト  | 禁止      | すべてのバージョンのVDA。VDA 7.xではレジストリで設定します |
| クライアントLPTポートリダイレクト  | 禁止      | すべてのバージョンのVDA。VDA 7.xではレジストリで設定します |

#### ICA/印刷

| 名前                       | デフォルト設定                    | VDA           |
|--------------------------|----------------------------|---------------|
| クライアントプリンターリダイレクト        | 許可                         | すべてのバージョンのVDA |
| デフォルトプリンター               | クライアントのメインプリンターをデフォルトに設定する | すべてのバージョンのVDA |
| プリンター割り当て                | ユーザーの現在のプリンター              | すべてのバージョンのVDA |
| プリンター自動作成イベントログの設定       | エラーおよび警告をログに記録する           | すべてのバージョンのVDA |
| セッションプリンター               | プリンターの指定なし                 | すべてのバージョンのVDA |
| プリンターの自動作成を待機する (デスクトップ) | 無効                         | すべてのバージョンのVDA |

#### ICA/印刷/クライアントプリンター

| 名前                   | デフォルト設定                           | VDA                             |
|----------------------|-----------------------------------|---------------------------------|
| クライアントプリンターを自動作成する   | すべてのクライアントプリンターを自動作成する            | すべてのバージョンのVDA                   |
| 汎用ユニバーサルプリンターを自動作成する | 無効                                | すべてのバージョンのVDA                   |
| クライアントプリンター名         | 標準のプリンター名                         | すべてのバージョンのVDA                   |
| プリントサーバーへの直接接続       | 有効                                | すべてのバージョンのVDA                   |
| プリンタードライバーのマッピングと互換性 | 規則の指定なし                           | すべてのバージョンのVDA                   |
| プリンタープロパティの保存        | クライアントに保存できない場合にのみユーザープロファイルに保存する | すべてのバージョンのVDA                   |
| クライアントプリンターの保持と復元    | 許可                                | VDA 5、5.5、および5.6 Feature Pack 1 |

#### ICA/印刷/ドライバー

| 名前                     | デフォルト設定                             | VDA           |
|------------------------|-------------------------------------|---------------|
| 付属のプリンタードライバーの自動インストール | 有効                                  | すべてのバージョンのVDA |
| ユニバーサルドライバーの優先度        | EMF; XPS; PCL5c; PCL4; PS           | すべてのバージョンのVDA |
| ユニバーサル印刷の使用            | 要求されたドライバーを使用できない場合にのみユニバーサル印刷を使用する | すべてのバージョンのVDA |

#### ICA/印刷/Universal Print Server

| 名前                                             | デフォルト設定 | VDA           |
|------------------------------------------------|---------|---------------|
| Universal Print Serverの有効化                     | 無効      | すべてのバージョンのVDA |
| Universal Print Server印刷データストリーム (CGP) ポート     | 7229    | すべてのバージョンのVDA |
| Universal Print Server入力データストリームの最大帯域幅 (Kbps)  | 0       | すべてのバージョンのVDA |
| Universal Print Server Webサービス (HTTP/SOAP) ポート | 8080    | すべてのバージョンのVDA |

## ICA/印刷/ユニバーサル印刷

| 名前               | デフォルト設定                                                                                                                                                                                                                                                 | VDA           |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| ユニバーサル印刷EMF処理モード | EMFスプールファイルを直接挿入する                                                                                                                                                                                                                                      | すべてのバージョンのVDA |
| ユニバーサル印刷イメージ圧縮制限 | 最高品質（無損失圧縮）                                                                                                                                                                                                                                             | すべてのバージョンのVDA |
| ユニバーサル印刷最適化デフォルト | イメージ圧縮 <ul style="list-style-type: none"> <li>必要なイメージ品質：標準品質</li> <li>ヘビーウェイト圧縮を有効にする：オフ</li> </ul> イメージおよびフォントのキャッシュ <ul style="list-style-type: none"> <li>埋め込みイメージのキャッシュを許可する：オン</li> <li>埋め込みフォントのキャッシュを許可する：オン</li> </ul> 非管理者によるこれらの設定の変更を許可する：オフ | すべてのバージョンのVDA |
| ユニバーサル印刷プレビューの設定 | 自動作成プリンターまたは汎用ユニバーサルプリンターの印刷プレビューを使用しない                                                                                                                                                                                                                 | すべてのバージョンのVDA |
| ユニバーサル印刷品質制限     | 制限なし                                                                                                                                                                                                                                                    | すべてのバージョンのVDA |

## ICA/セキュリティ

| 名前                 | デフォルト設定 | VDA                           |
|--------------------|---------|-------------------------------|
| SecureICAの最低暗号化レベル | 基本      | VDA for Windows Server OS 7以降 |

## ICA/サーバーの制限

| 名前               | デフォルト設定 | VDA                           |
|------------------|---------|-------------------------------|
| サーバーのアイドルタイマーの間隔 | 0ミリ秒    | VDA for Windows Server OS 7以降 |

## ICA/セッションの制限

| 名前          | デフォルト設定 | VDA                                                     |
|-------------|---------|---------------------------------------------------------|
| 切断セッションタイマー | 無効      | VDA 5、5.5、5.6 Feature Pack 1、VDA for Windows Desktop OS |

| 名前               | デフォルト設定 | VDA                                                                |
|------------------|---------|--------------------------------------------------------------------|
| 切断セッションタイマーの間隔   | 1440分   | 7以降<br>VDA 5、5.5、5.6 Feature Pack 1、VDA for Windows Desktop OS 7以降 |
| セッション接続タイマー      | 無効      | VDA 5、5.5、5.6 Feature Pack 1、VDA for Windows Desktop OS 7以降        |
| セッション接続タイマーの間隔   | 1440分   | VDA 5、5.5、5.6 Feature Pack 1、VDA for Windows Desktop OS 7以降        |
| セッションアイドルタイマー    | 有効      | VDA 5、5.5、5.6 Feature Pack 1、VDA for Windows Desktop OS 7以降        |
| セッションアイドルタイマーの間隔 | 1440分   | VDA 5、5.5、5.6 Feature Pack 1、VDA for Windows Desktop OS 7以降        |

### ICA/セッション画面の保持

| 名前                | デフォルト設定 | VDA           |
|-------------------|---------|---------------|
| セッション画面の保持        | 許可      | すべてのバージョンのVDA |
| セッション画面の保持のポート番号  | 2598    | すべてのバージョンのVDA |
| セッション画面の保持のタイムアウト | 180秒    | すべてのバージョンのVDA |

### ICA/タイムゾーン制御

| 名前                         | デフォルト設定          | VDA                           |
|----------------------------|------------------|-------------------------------|
| レガシークライアントのローカルタイムゾーンを検出する | 有効               | VDA for Windows Server OS 7以降 |
| クライアントのローカルタイムゾーンを使用する     | サーバーのタイムゾーンを使用する | すべてのバージョンのVDA                 |

### ICA/TWAINデバイス

| 名前                    | デフォルト設定 | VDA                                                                                        |
|-----------------------|---------|--------------------------------------------------------------------------------------------|
| クライアントTWAINデバイスリダイレクト | 許可      | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| TWAIN圧縮レベル            | 中       | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |

## ICA/USBデバイス

| 名前                           | デフォルト設定 | VDA                                                             |
|------------------------------|---------|-----------------------------------------------------------------|
| クライアントUSBデバイス最適化規則           | 規則の指定なし | VDA 7.6 FP3                                                     |
| クライアントUSBデバイスリダイレクト          | 禁止      | すべてのバージョンのVDA                                                   |
| クライアントUSBデバイスリダイレクト規則        | 規則の指定なし | すべてのバージョンのVDA                                                   |
| クライアントUSBプラグアンドプレイデバイスリダイレクト | 許可      | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |

## ICA/視覚表示

| 名前                | デフォルト設定             | VDA                                                             |
|-------------------|---------------------|-----------------------------------------------------------------|
| シンプルなグラフィック用の優先色数 | 24ビット/ピクセル          | VDA 7.6 FP3                                                     |
| ターゲットフレーム数        | 30fps               | すべてのバージョンのVDA                                                   |
| 表示品質              | 中                   | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| 圧縮にビデオコーデックを使用    | 使用可能な場合にビデオコーデックを使用 | VDA 7.6 FP3                                                     |

## ICA/視覚表示/動画

| 名前            | デフォルト設定        | VDA                                                                                        |
|---------------|----------------|--------------------------------------------------------------------------------------------|
| 画質の下限レベル      | 通常             | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| 動画圧縮          | 有効             | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| プログレッシブ圧縮のレベル | なし             | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| プログレッシブ圧縮     | 2147483647Kbps | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |

| 名前          | デフォルト設定 | VDA                                                                                        |
|-------------|---------|--------------------------------------------------------------------------------------------|
| のしきい値       | デフォルト設定 | VDA for Windows Desktop OS 7以降                                                             |
| 保持する最低フレーム数 | 10fps   | VDA 5.5、5.6 Feature Pack 1、VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |

#### ICA/視覚表示/静止画

| 名前           | デフォルト設定        | VDA           |
|--------------|----------------|---------------|
| エクストラ色圧縮     | 無効             | すべてのバージョンのVDA |
| エクストラ色圧縮しきい値 | 8192Kbps       | すべてのバージョンのVDA |
| ヘビーウェイト圧縮    | 無効             | すべてのバージョンのVDA |
| 非可逆圧縮のレベル    | 中              | すべてのバージョンのVDA |
| 非可逆圧縮のしきい値   | 2147483647Kbps | すべてのバージョンのVDA |

#### ICA/WebSocket

| 名前                      | デフォルト設定                               | VDA                                                             |
|-------------------------|---------------------------------------|-----------------------------------------------------------------|
| WebSocket接続             | 禁止                                    | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| WebSocketポート番号          | 8008                                  | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| WebSocket信頼される接続元サーバー一覧 | * (すべてのReceiver for WebサイトURLが信頼されます) | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |

#### 負荷管理

| 名前                    | デフォルト設定  | VDA                           |
|-----------------------|----------|-------------------------------|
| 同時ログオンセッション           | 2        | VDA for Windows Server OS 7以降 |
| CPU使用率                | 無効       | VDA for Windows Server OS 7以降 |
| CPU使用率から除外するプロセスの優先順位 | 通常以下および低 | VDA for Windows Server OS 7以降 |
| ディスク使用率               | 無効       | VDA for Windows Server OS 7以降 |

|                |                |                                      |
|----------------|----------------|--------------------------------------|
| 最大セッション数<br>名前 | 250<br>デフォルト設定 | VDA for Windows Server OS 7以降<br>VDA |
| メモリ使用率         | 無効             | VDA for Windows Server OS 7以降        |
| 基本メモリ使用量       | 負荷なし : 768MB   | VDA for Windows Server OS 7以降        |

#### Profile Management/上級設定

| 名前                         | デフォルト設定 | VDA           |
|----------------------------|---------|---------------|
| 自動構成を無効にする                 | 無効      | すべてのバージョンのVDA |
| 問題が発生する場合にユーザーをログオフ        | 無効      | すべてのバージョンのVDA |
| ロックされたファイルにアクセスする場合の試行数    | 5       | すべてのバージョンのVDA |
| ログオフ時のインターネットCookieファイルの処理 | 無効      | すべてのバージョンのVDA |

#### Profile Management/基本設定

| 名前                     | デフォルト設定                         | VDA           |
|------------------------|---------------------------------|---------------|
| アクティブライトバック            | 無効                              | すべてのバージョンのVDA |
| Profile managementの有効化 | 無効                              | すべてのバージョンのVDA |
| 除外グループ                 | 無効。(すべてのユーザーグループのプロファイルが処理されます) | すべてのバージョンのVDA |
| オフラインプロファイルサポート        | 無効                              | すべてのバージョンのVDA |
| ユーザーストアへのパス            | Windows                         | すべてのバージョンのVDA |
| ローカル管理者のログオン処理         | 無効                              | すべてのバージョンのVDA |
| 処理済みグループ               | 無効。(すべてのユーザーグループのプロファイルが処理されます) | すべてのバージョンのVDA |

#### Profile Management/クロスプラットフォーム設定

| 名前                       | デフォルト設定                                          | VDA           |
|--------------------------|--------------------------------------------------|---------------|
| クロスプラットフォーム設定ユーザーグループ    | 無効。( [処理済みグループ] で指定したすべてのユーザーグループのプロファイルが処理されます) | すべてのバージョンのVDA |
| クロスプラットフォーム設定の有効化        | 無効                                               | すべてのバージョンのVDA |
| クロスプラットフォーム定義へのパス        | 無効。(パスは指定されていません)                                | すべてのバージョンのVDA |
| クロスプラットフォーム設定ストアへのパス     | 無効。(Windows\PM_CMが使用されます)                        | すべてのバージョンのVDA |
| クロスプラットフォーム設定を作成するためのソース | 無効                                               | すべてのバージョンのVDA |

#### Profile Management/ファイルシステム/除外

| 名前             | デフォルト設定                          | VDA           |
|----------------|----------------------------------|---------------|
| 除外の一覧 - ディレクトリ | 無効。(ユーザープロファイルのすべてのフォルダーが同期されます) | すべてのバージョンのVDA |
| 除外の一覧 - ファイル   | 無効。(ユーザープロファイルのすべてのファイルが同期されません) | すべてのバージョンのVDA |

#### Profile Management/ファイルシステム/同期

| 名前            | デフォルト設定                     | VDA           |
|---------------|-----------------------------|---------------|
| 同期するディレクトリ    | 無効。(除外されていないフォルダーのみが同期されます) | すべてのバージョンのVDA |
| 同期するファイル      | 無効。(除外されていないファイルのみが同期されます)  | すべてのバージョンのVDA |
| ミラーリングするフォルダー | 無効。(フォルダーはミラーリングされません)      | すべてのバージョンのVDA |

#### Profile Management/フォルダーのリダイレクト

| 名前         | デフォルト設定 | VDA           |
|------------|---------|---------------|
| 管理者アクセスを許可 | 無効      | すべてのバージョンのVDA |
| ドメイン名を包含   | 無効      | すべてのバージョンのVDA |



| 名前                                                | デフォルト設定 | VDA |
|---------------------------------------------------|---------|-----|
| Profile Management/フォルダーのリダイレクト/AppData (Roaming) |         |     |

| 名前                          | デフォルト設定                                  | VDA           |
|-----------------------------|------------------------------------------|---------------|
| AppData (Roaming) パス        | 無効。(パスは指定されていません)                        | すべてのバージョンのVDA |
| AppData (Roaming) のリダイレクト設定 | [AppData (Roaming) パス] で指定したUNCパスにリダイレクト | すべてのバージョンのVDA |

#### Profile Management/フォルダーのリダイレクト/アドレス帳

| 名前             | デフォルト設定                     | VDA           |
|----------------|-----------------------------|---------------|
| アドレス帳パス        | 無効。(パスは指定されていません)           | すべてのバージョンのVDA |
| アドレス帳のリダイレクト設定 | [アドレス帳パス] で指定したUNCパスにリダイレクト | すべてのバージョンのVDA |

#### Profile Management/フォルダーのリダイレクト/デスクトップ

| 名前              | デフォルト設定                      | VDA           |
|-----------------|------------------------------|---------------|
| デスクトップパス        | 無効。(パスは指定されていません)            | すべてのバージョンのVDA |
| デスクトップのリダイレクト設定 | [デスクトップパス] で指定したUNCパスにリダイレクト | すべてのバージョンのVDA |

#### Profile Management/フォルダーのリダイレクト/ドキュメント

| 名前              | デフォルト設定                      | VDA           |
|-----------------|------------------------------|---------------|
| ドキュメントパス        | 無効。(パスは指定されていません)            | すべてのバージョンのVDA |
| ドキュメントのリダイレクト設定 | [ドキュメントパス] で指定したUNCパスにリダイレクト | すべてのバージョンのVDA |

#### Profile Management/フォルダーのリダイレクト/ダウンロード

| 名前              | デフォルト設定                      | VDA           |
|-----------------|------------------------------|---------------|
| ダウンロードパス        | 無効。(パスは指定されていません)            | すべてのバージョンのVDA |
| ダウンロードのリダイレクト設定 | [ダウンロードパス] で指定したUNCパスにリダイレクト | すべてのバージョンのVDA |

| 名前                                    | デフォルト設定                    | VDA           |
|---------------------------------------|----------------------------|---------------|
| Profile Management/フォルダーのリダイレクト/お気に入り | デフォルト設定                    | VDA           |
| お気に入りパス                               | 無効。(パスは指定されていません)          | すべてのバージョンのVDA |
| お気に入りのリダイレクト設定                        | [お気に入りパス]で指定したUNCパスにリダイレクト | すべてのバージョンのVDA |

Profile Management/フォルダーのリダイレクト/リンク

| 名前           | デフォルト設定                  | VDA           |
|--------------|--------------------------|---------------|
| リンクパス        | 無効。(パスは指定されていません)        | すべてのバージョンのVDA |
| リンクのリダイレクト設定 | [リンクパス]で指定したUNCパスにリダイレクト | すべてのバージョンのVDA |

Profile Management/フォルダーのリダイレクト/ミュージック

| 名前              | デフォルト設定                     | VDA           |
|-----------------|-----------------------------|---------------|
| ミュージックパス        | 無効。(パスは指定されていません)           | すべてのバージョンのVDA |
| ミュージックのリダイレクト設定 | [ミュージックパス]で指定したUNCパスにリダイレクト | すべてのバージョンのVDA |

Profile Management/フォルダーのリダイレクト/ピクチャ

| 名前            | デフォルト設定                   | VDA           |
|---------------|---------------------------|---------------|
| ピクチャパス        | 無効。(パスは指定されていません)         | すべてのバージョンのVDA |
| ピクチャのリダイレクト設定 | [ピクチャパス]で指定したUNCパスにリダイレクト | すべてのバージョンのVDA |

Profile Management/フォルダーのリダイレクト/保存したゲーム

| 名前               | デフォルト設定                      | VDA           |
|------------------|------------------------------|---------------|
| 保存したゲームパス        | 無効。(パスは指定されていません)            | すべてのバージョンのVDA |
| 保存したゲームのリダイレクト設定 | [保存したゲームパス]で指定したUNCパスにリダイレクト | すべてのバージョンのVDA |

Profile Management/フォルダーのリダイレクト/検索

| 名前          | デフォルト設定                  | VDA           |
|-------------|--------------------------|---------------|
| 検索パス        | 無効。(パスは指定されていません)        | すべてのバージョンのVDA |
| 検索のリダイレクト設定 | [検索パス] で指定したUNCパスにリダイレクト | すべてのバージョンのVDA |

Profile Management/フォルダーのリダイレクト/スタートメニュー

| 名前                | デフォルト設定                        | VDA           |
|-------------------|--------------------------------|---------------|
| スタートメニューパス        | 無効。(パスは指定されていません)              | すべてのバージョンのVDA |
| スタートメニューのリダイレクト設定 | [スタートメニューパス] で指定したUNCパスにリダイレクト | すべてのバージョンのVDA |

Profile Management/フォルダーのリダイレクト/ビデオ

| 名前           | デフォルト設定                   | VDA           |
|--------------|---------------------------|---------------|
| ビデオパス        | 無効。(パスは指定されていません)         | すべてのバージョンのVDA |
| ビデオのリダイレクト設定 | [ビデオパス] で指定したUNCパスにリダイレクト | すべてのバージョンのVDA |

Profile Management/ログ設定

| 名前                 | デフォルト設定 | VDA           |
|--------------------|---------|---------------|
| Active Directory操作 | 無効      | すべてのバージョンのVDA |
| 一般的な情報             | 無効      | すべてのバージョンのVDA |
| 一般的な警告             | 無効      | すべてのバージョンのVDA |
| ログの有効化             | 無効      | すべてのバージョンのVDA |
| ファイルシステム操作         | 無効      | すべてのバージョンのVDA |
| ファイルシステム通知         | 無効      | すべてのバージョンのVDA |

|                    |                                                                   |                        |
|--------------------|-------------------------------------------------------------------|------------------------|
| 名前オフ               | 無効<br>フォルト設定                                                      | VDA<br>でのバージョン<br>のVDA |
| ログオン               | 無効                                                                | すべてのバージョン<br>のVDA      |
| ログファイルの最大サイズ       | 1048576                                                           | すべてのバージョン<br>のVDA      |
| ログファイルへのパス         | 無効。<br>(%SystemRoot%\System32\Logfiles\UserProfileManagerに生成されます) | すべてのバージョン<br>のVDA      |
| 個人用ユーザー情報          | 無効                                                                | すべてのバージョン<br>のVDA      |
| ログオンおよびログオフ時のポリシー値 | 無効                                                                | すべてのバージョン<br>のVDA      |
| レジストリ操作            | 無効                                                                | すべてのバージョン<br>のVDA      |
| ログオフ時のレジストリ差分      | 無効                                                                | すべてのバージョン<br>のVDA      |

#### Profile Management/プロファイル制御

| 名前                          | デフォルト設定                                                            | VDA               |
|-----------------------------|--------------------------------------------------------------------|-------------------|
| キャッシュしたプロファイルを削除する前の待ち時間    | 0                                                                  | すべてのバージョン<br>のVDA |
| ログオフ時にローカルでキャッシュしたプロファイルの削除 | 無効                                                                 | すべてのバージョン<br>のVDA |
| ローカルプロファイル競合の制御             | ローカルプロファイルを使用                                                      | すべてのバージョン<br>のVDA |
| 既存のプロファイルの移行                | ローカルおよび移動                                                          | すべてのバージョン<br>のVDA |
| テンプレートプロファイルへのパス            | 無効。(ユーザーが最初にログオンするコンピューター上のデフォルトのユーザープロファイルから新しいユーザープロファイルが作成されます) | すべてのバージョン<br>のVDA |
| テンプレートプロファイルがローカル           | 無効                                                                 | すべての              |

| 名前                                          | デフォルト設定 | VDA<br>バージョンのVDA |
|---------------------------------------------|---------|------------------|
| プロファイルを上書きする                                | デフォルト設定 |                  |
| テンプレートプロファイルが移動プロファイルを上書きする                 | 無効      | すべてのバージョンのVDA    |
| すべてのログオンでCitrix固定プロファイルとして使用されるテンプレートプロファイル | 無効      | すべてのバージョンのVDA    |

#### Profile Management/レジストリ

| 名前    | デフォルト設定                                                 | VDA           |
|-------|---------------------------------------------------------|---------------|
| 除外の一覧 | 無効。(HKEY_CURRENT_USERハイブのすべてのレジストリキーがユーザーのログオフ時に処理されます) | すべてのバージョンのVDA |
| 包含の一覧 | 無効。(HKEY_CURRENT_USERハイブのすべてのレジストリキーがユーザーのログオフ時に処理されます) | すべてのバージョンのVDA |

#### Profile Management/ストリーム配信ユーザープロファイル

| 名前                      | デフォルト設定                        | VDA           |
|-------------------------|--------------------------------|---------------|
| 常時キャッシュ                 | 無効                             | すべてのバージョンのVDA |
| 常時キャッシュサイズ              | 0Mb                            | すべてのバージョンのVDA |
| プロファイルストリーム配信           | 無効                             | すべてのバージョンのVDA |
| ストリーム配信ユーザープロファイルグループ   | 無効。(OU内のすべてのユーザープロファイルが処理されます) | すべてのバージョンのVDA |
| 待機領域のロックファイルのタイムアウト(日数) | 1日                             | すべてのバージョンのVDA |

#### Receiver

| 名前                | デフォルト設定  | VDA                                                             |
|-------------------|----------|-----------------------------------------------------------------|
| StoreFrontアカウント一覧 | ストアの指定なし | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |

#### Virtual Delivery Agent

| 名前                   | デフォルト設定         | VDA                                                             |
|----------------------|-----------------|-----------------------------------------------------------------|
| コントローラー登録のIPv6ネットマスク | ネットマスクの指定なし     | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| コントローラー登録ポート         | 80              | すべてのバージョンのVDA                                                   |
| コントローラーSID           | SIDの指定なし        | すべてのバージョンのVDA                                                   |
| コントローラー              | Controllerの指定なし | すべてのバージョンのVDA                                                   |
| コントローラーの自動更新を有効にする   | 有効              | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| IPv6コントローラー登録のみを使用する | 無効              | VDA for Windows Server OS 7以降、およびVDA for Windows Desktop OS 7以降 |
| サイトGUID              | GUIDの指定なし       | すべてのバージョンのVDA                                                   |

#### 仮想IP

| 名前                | デフォルト設定 | VDA     |
|-------------------|---------|---------|
| 仮想IPループバックサポート    | 無効      | VDA 7.6 |
| 仮想IPループバックプログラム一覧 | なし      | VDA 7.6 |

#### HDX 3D Pro

| 名前              | デフォルト設定 | VDA                           |
|-----------------|---------|-------------------------------|
| 無損失を有効にする       | 有効      | VDA 5.5、および5.6 Feature Pack 1 |
| HDX 3D Pro品質レベル |         | VDA 5.5、および5.6 Feature Pack 1 |

# ポリシー設定リファレンス

Jan 04, 2017

ポリシーには、対象のセッションを制御するための設定項目（規則）を追加します。ここでは、その設定項目が依存するほかの設定項目や、関連する設定項目についても説明します。

## クイックリファレンス

次の各表は、ポリシーに追加できる設定の一覧です。これらの表では、左側の列がポリシーで制御するセッションの機能を示し、右側の列がその機能に対応する設定を示します。

## オーディオ

| 目的                         | 使用するポリシー設定        |
|----------------------------|-------------------|
| 複数オーディオデバイスの使用を制御する        | オーディオプラグアンドプレイ    |
| クライアント側のマイクからのオーディオ入力を制御する | クライアントマイクダイレクト    |
| クライアント側のオーディオの音質を制御する      | 音質                |
| クライアント側のスピーカーの使用を制御する      | クライアントオーディオリダイレクト |

## 帯域幅の制限

| 目的                              | 使用するポリシー設定                                                                                                             |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------|
| クライアントオーディオマッピングで使用される帯域幅を制限する  | <ul style="list-style-type: none"><li>オーディオリダイレクトの最大帯域幅 (Kbps) 、または</li><li>オーディオリダイレクトの最大帯域幅 (Kbps) (%)</li></ul>     |
| クリップボードマッピングで使用される帯域幅を制限する      | <ul style="list-style-type: none"><li>クリップボードリダイレクトの最大帯域幅 (Kbps) 、または</li><li>クリップボードリダイレクトの最大帯域幅 (Kbps) (%)</li></ul> |
| クライアント側ドライブへのアクセスで使用される帯域幅を制限する | <ul style="list-style-type: none"><li>ファイルリダイレクトの最大帯域幅 (Kbps) 、または</li><li>ファイルリダイレクトの最大帯域幅 (%)</li></ul>              |
| HDX MediaStreamマルチメディアアクセラレーション | <ul style="list-style-type: none"><li>HDX MediaStreamマルチメディアアクセラレーションの最大帯域幅 (Kbps) 、または</li></ul>                      |

| 目的                                  | 使用するポリシー設定                                                                                                             |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| クライアントセッションで使用される帯域幅を制限する           | DX MediaStream マルチメディアアクセラレーションの最大帯域幅 (%)                                                                              |
| 印刷                                  | <ul style="list-style-type: none"> <li>プリンターリダイレクトの最大帯域幅 (Kbps) 、または</li> <li>プリンターリダイレクトの最大帯域幅 (%)</li> </ul>         |
| カメラやスキャナーなどのTWAINデバイスで使用される帯域幅を制限する | <ul style="list-style-type: none"> <li>TWAINデバイスリダイレクトの最大帯域幅 (Kbps) 、または</li> <li>TWAINデバイスリダイレクトの最大帯域幅 (%)</li> </ul> |
| USBデバイスで使用される帯域幅を制限する               | <ul style="list-style-type: none"> <li>USBデバイスリダイレクトの最大帯域幅 (Kbps) 、または</li> <li>USBデバイスリダイレクトの最大帯域幅 (%)</li> </ul>     |

## クライアント側のドライブやデバイスのリダイレクト

| 目的                                 | 使用するポリシー設定                                                                                        |
|------------------------------------|---------------------------------------------------------------------------------------------------|
| ログオン時にクライアント側ドライブに接続する機能を制御する      | クライアントドライブに自動接続する                                                                                 |
| サーバーとローカルのクリップボード間でのデータ転送を制御する     | クライアントクリップボードリダイレクト                                                                               |
| クライアント側ドライブのマッピングを制御する             | クライアントドライブリダイレクト                                                                                  |
| クライアント側ハードディスクドライブの使用を制御する         | <ul style="list-style-type: none"> <li>クライアント側固定ドライブ、および</li> <li>クライアントドライブリダイレクト</li> </ul>     |
| クライアント側フロッピーディスクドライブの使用を制御する       | <ul style="list-style-type: none"> <li>クライアント側フロッピードライブ、および</li> <li>クライアントドライブリダイレクト</li> </ul>  |
| クライアント側ネットワークドライブの使用を制御する          | <ul style="list-style-type: none"> <li>クライアント側ネットワークドライブ、および</li> <li>クライアントドライブリダイレクト</li> </ul> |
| クライアント側CD、DVD、およびブルーレイドライブの使用を制御する | <ul style="list-style-type: none"> <li>クライアント側光学式ドライブ、および</li> <li>クライアントドライブリダイレクト</li> </ul>    |



|                                                               |                                                                                                          |
|---------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| 目的<br>クライアント側リムーバブルドライブの使用を制御する                               | 使用するポリシー設定<br>リムーバブルドライブ、および<br><ul style="list-style-type: none"> <li>クライアントドライブリダイレクト</li> </ul>       |
| デジタルカメラやスキャナーなどのクライアント側TWAINデバイスの使用および転送されるイメージデータの圧縮レベルを制御する | <ul style="list-style-type: none"> <li>クライアントTWAINデバイスリダイレクト</li> <li>TWAIN圧縮レベル</li> </ul>              |
| クライアント側USBデバイスの使用を制御する                                        | <ul style="list-style-type: none"> <li>クライアントUSBデバイスリダイレクト、および</li> <li>クライアントUSBデバイスリダイレクト規則</li> </ul> |
| WANを介した接続でのクライアント側ドライブへの書き込み速度を改善する                           | 非同期書き込みを使用する                                                                                             |

## コンテンツリダイレクト

|                                         |                     |
|-----------------------------------------|---------------------|
| 目的                                      | 使用するポリシー設定          |
| サーバーからユーザーデバイス側にコンテンツをリダイレクトするかどうかを制御する | ホストからクライアントへのリダイレクト |

## デスクトップUI

|                              |                     |
|------------------------------|---------------------|
| 目的                           | 使用するポリシー設定          |
| セッションでの壁紙の表示を制御する            | デスクトップの壁紙           |
| ウィンドウの内容を表示したままドラッグする機能を制御する | ドラッグ中にウィンドウの内容を表示する |

## グラフィック/マルチメディア

|                                              |            |
|----------------------------------------------|------------|
| 目的                                           | 使用するポリシー設定 |
| 仮想デスクトップがクライアント側に送信されるとき、1秒あたりのフレームの最大数を設定する | ターゲットフレーム数 |
| ユーザーデバイス側に表示されるイメージの表示品質を制御する                | 表示品質       |

| 目的                                                                         | 使用するポリシー設定                                                                                                                                                                                          |
|----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| セッションでのFlashコンテンツのレンダリングを制御する<br>セッションで特定のWebページ上のFlashコンテンツを表示するかどうかを制御する | Flashアクセラレーションのデフォルトの動作<br><ul style="list-style-type: none"> <li>Flashサーバー側でのコンテンツ取得URLリスト</li> <li>Flash URL互換性リスト</li> <li>Flashビデオフォールバック防止ポリシー設定</li> <li>Flashビデオフォールバック防止エラー*.swf</li> </ul> |

## マルチストリームネットワークトラフィックの優先度

| 目的                                                | 使用するポリシー設定                       |
|---------------------------------------------------|----------------------------------|
| マルチストリーム接続のICAトラフィックのポートを指定して、各ポートのネットワーク優先度を定義する | マルチポートポリシー                       |
| サーバーとユーザーデバイス間のマルチストリーム接続のサポートを有効にする              | マルチストリーム（コンピューターポリシーおよびユーザーポリシー） |

## 印刷

| 目的                                                                           | 使用するポリシー設定                                                                                          |
|------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| ログオン時のクライアントプリンターの自動作成を制御する                                                  | <ul style="list-style-type: none"> <li>クライアントプリンターを自動作成する、および</li> <li>クライアントプリンターリダイレクト</li> </ul> |
| プリンタープロパティの保存先を制御する                                                          | プリンタープロパティの保存                                                                                       |
| 印刷ジョブをサーバーから直接プリンターに送信するか、クライアント経由で送信するかを制御する                                | プリントサーバーへの直接接続                                                                                      |
| クライアント側プリンターの使用を制御する                                                         | クライアントプリンターリダイレクト                                                                                   |
| クライアントプリンターおよびネットワークプリンターの自動作成時に、Windowsに付属のプリンタードライバーを自動的にインストールするかどうかを制御する | 付属のプリンタードライバーの自動インストール                                                                              |
| ユニバーサルプリンタードライバーの使用を制御する                                                     | ユニバーサル印刷の使用                                                                                         |
| ローミングユーザーの接続方法に応じて自動作成されるプリンターを制御する                                          | デフォルトプリンター                                                                                          |

注：デスクトップまたはアプリケーションセッションでは、ポリシーを使用してスクリーンセーバーを有効にすることはできません。スクリーンセーバーが必要なユーザーの場合は、ユーザーデバイスにスクリーンセーバーを実装できます。

# ICAのポリシー設定

Jan 04, 2017

[ICA] カテゴリには、ICAリスナーの接続とクリップボードのマッピングに関する設定項目が含まれています。

## クライアントクリップボードリダイレクト

この設定項目では、クライアント側のクリップボードをサーバーのクリップボードにマップすることを許可または禁止します。

デフォルトでは許可されます。

これを禁止すると、セッションとローカルのクリップボード間でのデータ転送ができなくなります。ただし、セッション内で動作するアプリケーション間でのクリップボードを介したデータ転送は無効になりません。

この設定を許可したら、[クリップボードリダイレクトの最大帯域幅 (Kbps)] 設定または [クリップボードリダイレクトの最大帯域幅 (%) ] 設定を使用して、クライアント接続でクリップボードが使用できる帯域幅の上限を設定します。

## クライアントクリップボードに書き込みを許可する形式

[クライアントクリップボードの書き込み制限] 設定が有効な場合、ホスト側のクリップボードデータがクライアントエンドポイント側に共有されません。この [クライアントクリップボードに書き込みを許可する形式] 設定では、特定の種類のクリップボードデータの共有を許可します。これを行うには、この設定項目を有効にして、許可するデータ形式を追加します。

以下のシステム定義のクリップボードデータ形式を追加できます。

- CF\_TEXT
- CF\_BITMAP
- CF\_METAFILEPICT
- CF\_SYLK
- CF\_DIF
- CF\_TIFF
- CF\_OEMTEXT
- CF\_DIB
- CF\_PALETTE
- CF\_PENDATA
- CF\_RIFF
- CF\_WAVE
- CF\_UNICODETEXT
- CF\_ENHMETAFILE
- CF\_HDROP
- CF\_LOCALE
- CF\_DIBV5
- CF\_OWNERDISPLAY
- CF\_DSPTEXT
- CF\_DSPBITMAP
- CF\_DSPMETAFILEPICT
- CF\_DISPENHMETAFILE

また、以下のXenAppおよびXenDesktop用のカスタム定義のデータ形式を追加できます。

- CFX\_RICHTEXT
- CFX\_OfficeDrawingShape
- CFX\_BIFF8

カスタム定義のデータ形式を追加することもできます。この場合、データ形式の名前がシステムに登録されたものと一致する必要があります。また、形式名の太文字と小文字は区別されます。

[クライアントクリップボードリダイレクト]設定または[クライアントクリップボードの書き込み制限]設定で[禁止]または[無効]が設定されている場合、この設定項目は無視されます。

### デスクトップの起動

この設定により、VDAのDirect Access Usersグループの非管理者ユーザーによるICAコネクションを使ったVDA上のセッションへの接続を許可または禁止します。

デフォルトでは、管理者以外のユーザーはこれらのセッションに接続できません。

この設定は、RDP接続を使用しているVDAのDirect Access Usersグループの非管理者ユーザーには影響がありません。これらのユーザーは、この設定が有効または無効になっているかに関わらずVDAに接続できます。この設定は、VDAのDirect Access Usersグループではない非管理者ユーザーには影響がありません。これらのユーザーは、この設定が有効または無効になっているかに関わらずVDAに接続できます。

### ICAリスナー接続タイムアウト

注：この設定は、Virtual Delivery Agent 5.0、5.5、および5.6 Feature Pack 1にのみ適用されます。

この設定項目では、ICAプロトコルによる接続が完了するまでの最大待機時間を指定します。

デフォルトの最大待機時間は、120000ミリ秒（2分）です。

### ICAリスナーポートの番号

この設定では、サーバー上のICAプロトコルで使用されるTCP/IPポートを指定します。

デフォルトのポート番号は、1494に設定されています。

ほかのポートを指定する場合は、0から65535の、ほかの既知のポートと競合しない番号を使用してください。変更したポート番号を有効にするには、サーバーを再起動する必要があります。サーバー上のポート番号を変更した場合は、そのサーバーに接続するReceiverやプラグインソフトウェア側でもポート番号を変更する必要があります。

### クライアント接続での非公開アプリケーションの起動

この設定項目では、サーバー上のリモートデスクトップを介した開始アプリケーションの起動を許可するかどうかを指定します。

デフォルトでは、サーバー上のリモートデスクトップを介した開始アプリケーションの機能は許可されません。

### クライアントクリップボードの書き込み制限

この設定項目を有効にすると、ホスト側のクリップボードデータがクライアントエンドポイント側に共有されなくなります。この場合、特定のデータの共有を許可するには、[クライアントクリップボードに書き込みを許可する形式]設定を使用します。

デフォルトでは、無効に設定されています。

## セッションクリップボードの書き込み制限

この設定項目を有効にすると、クライアント側のクリップボードデータがユーザーセッション側に共有されなくなります。この場合、特定のデータの共有を許可するには、[セッションクリップボードに書き込みを許可する形式] 設定を使用します。

デフォルトでは、無効に設定されています。

### セッションクリップボードに書き込みを許可する形式

[セッションクリップボードの書き込み制限] 設定が有効な場合、クライアント側のクリップボードデータがセッション内のアプリケーション側に共有されません。この [セッションクリップボードに書き込みを許可する形式] 設定では、特定の種類のクリップボードデータの共有を許可します。

以下のシステム定義のクリップボードデータ形式を追加できます。

- CF\_TEXT
- CF\_BITMAP
- CF\_METAFILEPICT
- CF\_SYLK
- CF\_DIF
- CF\_TIFF
- CF\_OEMTEXT
- CF\_DIB
- CF\_PALETTE
- CF\_PENDATA
- CF\_RIFF
- CF\_WAVE
- CF\_UNICODETEXT
- CF\_ENHMETAFILE
- CF\_HDROP
- CF\_LOCALE
- CF\_DIBV5
- CF\_OWNERDISPLAY
- CF\_DSPTEXT
- CF\_DSPBITMAP
- CF\_DSPMETAFILEPICT
- CF\_DISPENHMETAFILE

また、以下のXenAppおよびXenDesktop用のカスタム定義のデータ形式を追加できます。

- CFX\_RICHTEXT
- CFX\_OfficeDrawingShape
- CFX\_BIFF8

カスタム定義のデータ形式を追加することもできます。この場合、データ形式の名前がシステムに登録されたものと一致する必要があります。また、形式名の太文字と小文字は区別されます。

[クライアントクリップボードリダイレクト] 設定または [セッションクリップボードの書き込み制限] 設定で [禁止] または [無効] が設定されている場合、この設定項目は無視されます。

# クライアントの自動再接続のポリシー設定

Jan 04, 2017

[クライアントの自動再接続] カテゴリには、セッションの自動再接続の制御に関する設定項目が含まれています。

## クライアントの自動再接続

この設定項目では、接続が中断した後で同じクライアントから自動再接続することを許可または禁止します。

デフォルトでは許可されます。

自動再接続を許可すると、ユーザーはセッションが切断された時点の状態に戻って作業を再開できます。自動再接続機能では、切断された接続が検出されてそのセッションにユーザーが再接続されます。

ただし、ユーザーのセッションIDと資格情報を保持するCookieが使用されない場合は、既存のセッションに再接続できず、新しいセッションが開始されることがあります。再接続に時間がかかってCookieの有効期限が切れた場合や、ユーザーが資格情報を入力する必要がある場合には、Cookieは使用されません。また、ユーザーが意図的に切断したセッションには再接続されません。

アプリケーションセッションでは、自動再接続が許可されている場合、Receiverは、接続に成功するかユーザーがキャンセルするまで再接続を繰り返し試行します。

デスクトップセッションでは、自動再接続が許可されている場合、Receiverは、接続に成功するかユーザーがキャンセルするまで、指定された期間の間、再接続を繰り返し試行します。デフォルトでは、この期間は5分です。この期間を変更するには、ユーザーデバイスで以下のレジストリを編集します。

`HKLM\Software\Citrix\ICA Client\TransportReconnectRetryMaxTimeSeconds; DWORD;<seconds>`

ここで、<seconds>は秒数です。この期間が経過するとセッションへの再接続が試行されなくなります。

## クライアントの自動再接続時の認証

この設定項目では、自動再接続時に認証処理を必要とするかどうかを制御します。[認証を必要とする]を選択すると、クライアントの自動再接続時に認証のためのダイアログボックスが開きます。

デフォルトでは、認証は要求されません。

ユーザーが最初にセッションにログオンすると、そのユーザーの資格情報は暗号化されてメモリ上に格納され、その暗号キーを含んだCookieが作成され、Receiverに送信されます。この設定項目を構成して[認証を必要とする]を選択すると、Cookieは使用されなくなります。その代わりに、切断セッションへの再接続時に、ユーザーの資格情報を入力するためのダイアログボックスが開きます。

## クライアントの自動再接続のログ

この設定では、クライアントの自動再接続イベントをログに記録するかどうかを制御します。

デフォルトでは、ログは無効になっています。

ログを有効にすると、サーバーのシステムログに自動再接続の成功および失敗イベントが記録されます。これらのイベントは、そのイベントが発生した個々のサーバーのシステムログに記録されます。

# オーディオのポリシー設定

Jan 04, 2017

[オーディオ] カテゴリには、ユーザーデバイスがパフォーマンスを低下させずにセッションでオーディオを送受信することを許可するための設定項目が含まれています。

## UDPでのオーディオリアルタイムトランスポート

この設定項目では、ホストとユーザーデバイス間のユーザーデータグラムプロトコル (UDP) を使用したオーディオリアルタイムトランスポート (RTP) でのオーディオ転送を許可または禁止します。この設定項目を無効にすると、オーディオがTCF上で送受信されます。

デフォルトでは許可されます。

## オーディオプラグアンドプレイ

この設定項目では、録音やサウンド再生のための複数のオーディオデバイスの使用を許可または禁止します。

デフォルトでは許可されます。

この設定項目は、WindowsサーバーOSマシンのみ適用されます。

## 音質

この設定項目では、ユーザーセッション内で受信されるサウンドの品質を指定します。

デフォルトでは、[高 - 高品位オーディオ] が指定されています。

音質を制御するには、次のオプションから1つを選択します。

- 狭帯域接続には [低 - 低速接続用] を選択します。この設定では、サウンドデータが最大16Kbpsまで圧縮されてから転送されます。圧縮により、再生または録音される音質は著しく低下しますが、モデム接続などの狭帯域幅接続に最適です。
- VoIP (Voice over IP) アプリケーションを配信する場合、512Kbps未満の低速なネットワーク接続回線でメディアアプリケーションを配信する場合、または輻輳やパケット損失が生じる環境では、[中 - スピーチに最適化] を選択します。高速にエンコーディングされるため、ソフトフォンや統合コミュニケーションアプリケーションなどのメディア処理をサーバー側で行う場合に適しています。

この音質レベルでは、オーディオデータが最大64Kbpsまで圧縮されてからユーザーデバイスに転送されます。この圧縮により、ユーザーデバイス上でのオーディオ再生の品質はやや低下しますが、遅延は少なくなり、帯域幅の消費も少なくなります。VoIPアプリケーションで十分な音質が得られない場合は、[UDPでのオーディオリアルタイムトランスポート] 設定で [許可] を選択します。

- 帯域幅が十分で、サウンドの音質が重要である場合は、[高 - 高品位オーディオ] を選択します。この設定では、ネイティブのサウンドデータを再生または録音できます。サウンドデータは、CDレベルの音質が維持される112Kbpsの高品質レベルで圧縮されます。ただし、大量のネットワークデータ転送が要求されるため、CPUおよびネットワークに負担がかかる場合があります。

録音と再生を同時に行った場合、消費帯域幅は2倍になります。

帯域幅の上限を指定するには、[オーディオリダイレクトの最大帯域幅 (Kbps)] 設定または [オーディオリダイレクトの最大帯域幅 (%)] 設定を使用します。

## クライアントオーディオリダイレクト



この設定項目では、サーバーでホストされているアプリケーションから、ユーザーデバイスにインストールされているサウンドデバイスを介してサウンドを再生したり、オーディオ入力を録音したりすることを許可または禁止します。

デフォルトでは許可されます。

この設定を許可したら、オーディオの再生や録音で使用できる帯域幅の上限を設定します。これにより、アプリケーションのパフォーマンスが向上しますが、音質が低下することがあります。録音と再生を同時に行った場合、消費帯域幅は2倍になります。帯域幅の上限を指定するには、[オーディオリダイレクトの最大帯域幅 (Kbps)] 設定または [オーディオリダイレクトの最大帯域幅 (%)] 設定を使用します。

WindowsサーバーOSマシンで複数のオーディオデバイスをサポートするには、[オーディオプラグアンドプレイ] 設定で [有効] が選択されていることも確認してください。

**重要：** [クライアントオーディオリダイレクト] 設定で [禁止] を選択すると、すべてのHDXオーディオ機能が無効になります。

#### クライアントマイクリダイレクト

この設定項目では、クライアント側のマイクのリダイレクトを有効または無効にします。この設定を有効にすると、セッション内でクライアント側のマイクを使ってオーディオを録音できるようになります。

デフォルトでは許可されます。

セキュリティの設定により、ユーザーデバイスに信頼されていないサーバーからユーザーデバイス側のマイクにアクセスしようとするときに、警告メッセージが表示されます。ユーザーは、このメッセージに対してアクセスを許可したり拒否したりできます。この警告は、ユーザーがCitrix Receiver側で無効にできます。

WindowsサーバーOSマシンで複数のオーディオデバイスをサポートするには、[オーディオプラグアンドプレイ] 設定で [有効] が選択されていることも確認してください。

ユーザーデバイス側で [クライアントオーディオリダイレクト] 設定が無効になっている場合、この設定項目は無視されません。

# 帯域幅のポリシー設定

Jan 04, 2017

[帯域幅] カテゴリには、クライアントセッションでの消費帯域幅に関する問題を避けるための設定項目が含まれています。  
重要：これらの設定項目を [マルチストリーム] 設定と一緒に使用すると、意図したとおりに動作しなくなる場合があります。ポリシーで [マルチストリーム] 設定を使用する場合は、帯域幅を制限する設定項目を追加しないようにしてください。  
オーディオリダイレクトの最大帯域幅 (Kbps)

この設定項目では、クライアント側デバイスによるオーディオの再生や録音で使用可能な最大帯域幅を、キロビット/秒 (Kbps) 単位で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [オーディオリダイレクトの最大帯域幅 (%) ] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

## オーディオリダイレクトの最大帯域幅 (%)

この設定項目では、クライアント側デバイスによるオーディオの再生や録音で使用可能な最大帯域幅を、セッション全体に占める割合で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [オーディオリダイレクトの最大帯域幅 (Kbps) ] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

この設定を使用する場合は、セッションで使用可能な総帯域幅の最大値を指定する [セッション全体の最大帯域幅] 設定も使用する必要があります。

## USBデバイスリダイレクトの最大帯域幅 (Kbps)

この設定項目では、クライアント側のUSBデバイスにアクセスするときに使用可能な最大帯域幅を、キロビット/秒 (Kbps) 単位で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [USBデバイスリダイレクトの最大帯域幅 (%) ] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

## USBデバイスリダイレクトの最大帯域幅 (%)

この設定項目では、クライアント側のUSBデバイスにアクセスするときに使用可能な最大帯域幅を、セッション全体に対する割合で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [USBデバイスリダイレクトの最大帯域幅 (Kbps) ] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

この設定を使用する場合は、セッションで使用可能な総帯域幅の最大値を指定する [セッション全体の最大帯域幅] 設定も使用する必要があります。

## クリップボードリダイレクトの最大帯域幅 (Kbps)

この設定項目では、セッションとローカルのクリップボード間でのデータ転送で使用可能な最大帯域幅を、キロビット/秒 (Kbps) 単位で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [クリップボードリダイレクトの最大帯域幅 (%) ] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

#### クリップボードリダイレクトの最大帯域幅 (%)

この設定項目では、セッションとローカルのクリップボード間でのデータ転送で使用可能な最大帯域幅を、セッション全体に対する割合で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [クリップボードリダイレクトの最大帯域幅 (Kbps) ] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

この設定を使用する場合は、セッションで使用可能な総帯域幅の最大値を指定する [セッション全体の最大帯域幅] 設定も使用する必要があります。

#### COMポートリダイレクトの最大帯域幅 (Kbps)

注: Virtual Delivery Agent 7.xでは、レジストリを使用してこの機能を構成します (「[レジストリを使ったCOMポートおよびLPTポートリダイレクト設定の構成](#)」参照)。

この設定項目では、クライアント側COMポートにアクセスするときに使用可能な最大帯域幅を、キロビット/秒 (Kbps) 単位で指定します。この設定および [COMポートリダイレクトの最大帯域幅 (%) ] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

#### COMポートリダイレクトの最大帯域幅 (%)

注: Virtual Delivery Agent 7.xでは、レジストリを使用してこの機能を構成します (「[レジストリを使ったCOMポートおよびLPTポートリダイレクト設定の構成](#)」参照)。

この設定項目では、クライアント側COMポートにアクセスするときに使用可能な最大帯域幅を、セッション全体に対する割合で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [COMポートリダイレクトの最大帯域幅 (Kbps) ] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

この設定を使用する場合は、セッションで使用可能な総帯域幅の最大値を指定する [セッション全体の最大帯域幅] 設定も使用する必要があります。

#### ファイルリダイレクトの最大帯域幅 (Kbps)

この設定項目では、クライアント側ドライブにアクセスするときに使用可能な最大帯域幅を、キロビット/秒 (Kbps) 単位で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [ファイルリダイレクトの最大帯域幅 (%) ] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

## ファイルリダイレクトの最大帯域幅 (%)

この設定項目では、クライアント側ドライブにアクセスするときに使用可能な最大帯域幅を、セッション全体に対する割合で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [ファイルリダイレクトの最大帯域幅 (Kbps)] の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

この設定を使用する場合は、セッションで使用可能な総帯域幅の最大値を指定する [セッション全体の最大帯域幅] 設定も使用する必要があります。

## HDX MediaStreamマルチメディアアクセラレーションの最大帯域幅 (Kbps)

この設定項目では、HDX MediaStreamマルチメディアアクセラレーションによりストリーム配信されるオーディオやビデオで使用可能な最大帯域幅を、キロビット/秒 (Kbps) 単位で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [HDX MediaStreamマルチメディアアクセラレーションの最大帯域幅 (%)] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

## HDX MediaStreamマルチメディアアクセラレーションの最大帯域幅 (%)

この設定項目では、HDX MediaStreamマルチメディアアクセラレーションによりストリーム配信されるオーディオやビデオで使用可能な最大帯域幅を、セッション全体に対する割合で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [HDX MediaStreamマルチメディアアクセラレーションの最大帯域幅 (Kbps)] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

この設定を使用する場合は、セッションで使用可能な総帯域幅の最大値を指定する [セッション全体の最大帯域幅] 設定も使用する必要があります。

## LPTポートリダイレクトの最大帯域幅 (Kbps)

注: Virtual Delivery Agent 7.xでは、レジストリを使用してこの機能を構成します (「[レジストリを使ったCOMポートおよびLPTポートリダイレクト設定の構成](#)」参照)。

この設定項目では、クライアント側LPTポートを使用する印刷ジョブで使用可能な最大帯域幅を、キロビット/秒 (Kbps) 単位で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [LPTポートリダイレクトの最大帯域幅 (%)] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

## LPTポートリダイレクトの最大帯域幅 (%)

注: Virtual Delivery Agent 7.xでは、レジストリを使用してこの機能を構成します (「[レジストリを使ったCOMポートおよびLPTポートリダイレクト設定の構成](#)」参照)。

この設定項目では、クライアント側LPTポートを使用する印刷ジョブで使用可能な最大帯域幅を、セッション全体に対する割合で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [LPTポートリダイレクトの最大帯域幅 (Kbps)] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

この設定を使用する場合は、セッションで使用可能な総帯域幅の最大値を指定する [セッション全体の最大帯域幅] 設定も使用する必要があります。

#### セッション全体の最大帯域幅

この設定項目では、セッションで使用可能な総帯域幅の最大値を、キロビット/秒 (Kbps) 単位で指定します。

適用できる帯域幅の上限は、10Mbps (10,000Kbps) です。デフォルトでは、上限なし (0) が指定されています。

狭帯域幅接続で、セッションでの使用帯域幅が原因でほかのアプリケーションでのデータ転送パフォーマンスが低下する場合に、この設定項目を使用します。

#### プリンターリダイレクトの最大帯域幅 (Kbps)

この設定項目では、クライアント側プリンターにアクセスするときに使用可能な最大帯域幅を、キロビット/秒 (Kbps) 単位で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [プリンターリダイレクトの最大帯域幅 (%) ] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

#### プリンターリダイレクトの最大帯域幅 (%)

この設定項目では、クライアント側プリンターにアクセスするときに使用可能な最大帯域幅を、セッション全体に対する割合で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [プリンターリダイレクトの最大帯域幅 (Kbps)] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

この設定を使用する場合は、セッションで使用可能な総帯域幅の最大値を指定する [セッション全体の最大帯域幅] 設定も使用する必要があります。

#### TWAINデバイスリダイレクトの最大帯域幅 (Kbps)

この設定項目では、クライアント側TWAINデバイスにアクセスするときに使用可能な最大帯域幅を、キロビット/秒 (Kbps) 単位で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [TWAINデバイスリダイレクトの最大帯域幅 (%) ] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

#### TWAINデバイスリダイレクトの最大帯域幅 (%)

この設定項目では、クライアント側TWAINデバイスにアクセスするときに使用可能な最大帯域幅を、セッション全体に対する割合で指定します。

デフォルトでは、上限なし (0) が指定されています。

この設定および [TWAINデバイスリダイレクトの最大帯域幅 (Kbps) ] 設定の両方を指定した場合、より高い制限 (より小さい値) の設定が適用されます。

この設定を使用する場合は、セッションで使用可能な総帯域幅の最大値を指定する [セッション全体の最大帯域幅] 設定も使用する必要があります。

# クライアントセンサーのポリシー設定

Jan 04, 2017

[クライアントセンサー] カテゴリには、ユーザーセッションでのモバイルデバイスのセンサー情報の制御に関する設定項目が含まれています。

## クライアントデバイスの位置情報をアプリケーションで使用する

この設定項目では、セッション内のアプリケーションをモバイルデバイス上で使用する場合に、そのモバイルデバイスの位置情報をアプリケーションで使用するのを許可または禁止します。

デフォルトでは、禁止されます。

位置情報の使用が禁止されている場合、アプリケーションからの位置情報の取得要求に対して「アクセス拒否」が返されません。

位置情報の使用が許可されている場合でも、Receiverからの位置情報の要求を拒否することで、位置情報の使用をユーザーが拒否できます。AndroidおよびiOSデバイスでは、そのセッションで最初に位置情報への要求が発生したときにメッセージが表示されます。

[クライアントデバイスの位置情報をアプリケーションで使用する]設定をサポートするアプリケーションを開発する場合は、以下の点に注意してください。

- 位置情報が常に使用可能であるとは限りません。これは、以下の理由によります。
  - 位置情報の使用をユーザーが拒否する可能性がある。
  - アプリケーションを実行している間に位置情報が提供されない、または位置が変化する可能性がある。
  - 位置情報の提供をサポートしないほかのデバイスからアプリケーションに再接続する可能性がある。
- 位置情報をサポートするアプリケーションの設定として、以下の点を考慮してください。
  - 位置情報の使用をデフォルトで無効にする。
  - アプリケーションの実行時にユーザーが位置情報の使用を許可したり禁止したりできる。
  - アプリケーションでキャッシュされた位置情報データをユーザーが消去できる（ただしReceiverは位置情報データをキャッシュしません）。
- そのアプリケーションでの目的に適したデータが取得されるように、位置情報の精度を管理できなければなりません。また、位置情報の使用について、すべての関連法域の法規に準拠しなければなりません。
- 位置情報を使用するときは、保護された接続（SSL/TLSやVPNによる接続など）が使用されるようにします。Citrix Receiverで接続するサーバーは、信頼されたものである必要があります。
- 位置情報サービスの使用に関して法的なアドバイスを得ることを検討してください。

# デスクトップUIのポリシー設定

Jan 04, 2017

[デスクトップUI] カテゴリには、クライアント接続での視覚効果を制御して使用帯域幅を管理するための設定項目が含まれています。視覚効果に含まれるのは、デスクトップの壁紙、メニューのアニメーション、およびドラッグ中にウィンドウの内容を表示する機能です。WANなどの狭帯域幅接続で視覚効果を無効にすると、公開アプリケーションのパフォーマンスが向上します。

## デスクトップコンポジションリダイレクト

この設定では、ローカルのDirectXグラフィック処理をユーザーデバイス側のGPU (Graphics Processing Unit) またはIGP (Integrated Graphics Processor) で行い、より滑らかなWindowsデスクトップ操作を提供するかどうかを指定します。デスクトップコンポジションリダイレクトを有効にすると、Windowsデスクトップの操作レスポンスが向上し、サーバーの高いスケーラビリティが維持されます。

デフォルトでは、デスクトップコンポジションのリダイレクトは、無効になっているか (7.6 FP3~最新) 有効になっていません (5.6~7.6 FP2)。

デスクトップコンポジションリダイレクトを無効にしてユーザーセッションに必要な帯域幅を減らすには、この設定項目で [無効] を選択します。

## デスクトップコンポジションの画質

この設定項目では、デスクトップコンポジションリダイレクトで使用される画質を指定します。

デフォルトでは、[高] に設定されています。

[高]、[中]、[低]、または [無損失] から選択します。

## デスクトップの壁紙

この設定項目では、ユーザーセッションでの壁紙の表示を許可または禁止します。

デフォルトでは、ユーザーセッションで壁紙を表示できます。

デスクトップの壁紙を非表示にしてユーザーセッションに必要な帯域幅を減らすには、ポリシーにこの設定を追加して [禁止] をクリックします。

## メニューをアニメーション化する

この設定項目では、ユーザーセッションでのメニューアニメーションを許可または禁止します。

デフォルトでは許可されます。

メニューアニメーションは、アクセスを簡単にするためのMicrosoftの個人優先設定です。これが有効な場合、スクロールまたはフェードインすることでメニューが表示されるのが少し遅れることとなります。矢印アイコンはメニュー下部に表示されます。そのアイコン上にマウスポインターを置くと、メニューの内容が表示されます。

この設定項目が [許可] に設定されている場合、デスクトップでメニューアニメーションが有効で、またメニューアニメーションMicrosoft個人優先設定が有効です。

注: メニューアニメーションMicrosoft個人優先設定への変更がデスクトップに変更されます。これにより、セッションの終了時に変更を破棄するようデスクトップが設定されます。セッションでメニューアニメーションを有効にしたユーザーは、



デスクトップ上の以降のセッションではメニューアニメーションを使用できない可能性があります。メニューアニメーションが必要なユーザーについては、デスクトップのマスターイメージのMicrosoft設定を有効にするか、またはデスクトップでユーザーの変更を維持する必要があります。

#### ドラッグ中にウィンドウの内容を表示する

この設定項目では、ウィンドウをドラッグするときにウィンドウの内容を表示する機能を許可または禁止します。

デフォルトでは許可されます。

[許可] を選択すると、ウィンドウをドラッグするときに内容が表示されたままになります。[禁止] を選択すると、ドロップするまでウィンドウの外枠のみが表示されます。

# エンドユーザーモニタリングのポリシー設定

Jan 04, 2017

[エンドユーザーモニタリング] カテゴリには、セッショントラフィックの測定に関する設定項目が含まれています。

## ICA往復測定

この設定項目では、アクティブな接続に対してICA往復測定を実行するかどうかを決定します。

デフォルトでは、ICA往復測定が実行されます。

デフォルトでは、ユーザーの操作によるいくつかのトラフィックが発生するまで、ICA往復測定の開始は遅延されます。このため、ユーザーが操作していないにもかかわらずICA往復測定によるICAトラフィックが発生することはありません。

## ICA往復測定間隔

この設定項目では、ICA往復測定を実行する頻度を秒単位で指定します。

デフォルトでは、15秒ごとに測定が実行されます。

## アイドル接続のICA往復測定

この設定項目では、アイドル状態の接続に対してICA往復測定を実行するかどうかを決定します。

デフォルトでは、アイドル接続に対してICA往復測定は実行されません。

デフォルトでは、ユーザーの操作によるいくつかのトラフィックが発生するまで、ICA往復測定の開始は遅延されます。このため、ユーザーが操作していないにもかかわらずICA往復測定によるICAトラフィックが発生することはありません。

# デスクトップエクスペリエンス拡張のポリシー設定

Jan 04, 2017

この設定項目では、ローカルでWindows 7デスクトップを実行しているユーザーに対して、サーバーオペレーティングシステム上のセッションにWindows 7デスクトップテーマを適用するかどうかを構成します。

デフォルトでは、許可されています。

Windowsクラシックテーマが選択されたユーザープロファイルが存在する仮想デスクトップでは、この設定項目を有効にしてもデスクトップエクスペリエンス拡張が提供されません。この設定項目が未構成または無効の場合、Windows 7テーマのユーザーがWindows Server 2012上の仮想デスクトップにログオンすると、テーマの適用に失敗したことを示すエラーメッセージが表示されます。

これらの問題は、ユーザープロファイルをリセットすることで解決されます。

実行中のユーザーセッションが存在する仮想デスクトップでこの設定項目を有効から無効に変更すると、Windows 7テーマおよびWindowsクラシックテーマでの表示に問題が発生します。この問題を避けるには、この設定項目の構成を変更した後で仮想デスクトップを再起動してください。また、管理者は仮想デスクトップの移動プロファイルを削除する必要もあります。さらに、プロファイル間の一貫性の問題を避けるため、仮想デスクトップのほかのユーザープロファイルもすべて削除することをお勧めします。

移動プロファイルが使用される環境では、プロファイルを共有するすべての仮想デスクトップでデスクトップエクスペリエンス拡張機能の有効/無効を統一してください。

サーバーOSを実行する仮想デスクトップとクライアントOSを実行する仮想デスクトップで移動プロファイルを共有することは推奨されません。サーバーOSとクライアントOSのプロファイルは異なるため、移動プロファイルを共有するとプロファイル内のプロパティの整合性に問題が生じることがあります。

# ファイルリダイレクトのポリシー設定

Jan 04, 2017

[ファイルリダイレクト] カテゴリには、クライアント側ドライブのマッピングと最適化に関する設定項目が含まれています。

## クライアントドライブに自動接続する

この設定項目では、ログオン時にクライアント側のドライブに自動的にマップすることを許可または禁止します。

デフォルトでは許可されます。

この設定項目をポリシーに追加する場合は、ドライブの種類別の設定項目についても確認してください。たとえば、クライアント側のCD-ROMドライブへの自動接続を許可するには、この設定および [クライアント側光学式ドライブ] 設定を許可します。

関連する設定項目は以下のとおりです。

- クライアントドライブリダイレクト
- クライアント側フロッピードライブ
- クライアント側光学式ドライブ
- クライアント側固定ドライブ
- クライアント側ネットワークドライブ
- クライアント側リムーバブルドライブ

## クライアントドライブリダイレクト

この設定項目では、ファイルのクライアント側ドライブへのリダイレクトおよびクライアント側ドライブからのリダイレクトを有効または無効にします。

デフォルトでは有効になっています。

この設定を有効にすると、ユーザーはクライアント側のすべてのドライブにファイルを保存できるようになります。この設定を無効にすると、すべてのクライアント側ドライブにファイルを保存できなくなります。このとき、[クライアント側フロッピードライブ] 設定や [クライアント側ネットワークドライブ] 設定などの個々のファイルリダイレクト設定の内容は考慮されません。

関連する設定項目は以下のとおりです。

- クライアント側フロッピードライブ
- クライアント側光学式ドライブ
- クライアント側固定ドライブ
- クライアント側ネットワークドライブ
- クライアント側リムーバブルドライブ

## クライアント側固定ドライブ

この設定項目では、クライアント側の固定ドライブにアクセスしたりファイルを保存したりすることを許可または禁止します。

デフォルトでは許可されます。

この設定項目をポリシーに追加するときは、[クライアントドライブリダイレクト] 設定で [許可] が選択されていることを

確認してください。この設定が無効な場合、クライアント側固定ドライブはマップされず、ユーザーが手作業でそれらのドライブにアクセスすることもできなくなります。このとき、[クライアント側固定ドライブ]設定の内容は考慮されません。

ユーザーがログオンしたときに固定ドライブに自動接続できるようにするには、[クライアントドライブに自動接続する]設定を使用します。

### クライアント側フロッピードライブ

この設定項目では、クライアント側のフロッピードライブにアクセスしたりファイルを保存したりすることを許可または禁止します。

デフォルトでは許可されます。

この設定項目をポリシーに追加するときは、[クライアントドライブリダイレクト]設定で[許可]が選択されていることを確認してください。この設定が無効な場合、クライアント側フロッピードライブはマップされず、ユーザーが手作業でそれらのドライブにアクセスすることもできなくなります。このとき、[クライアント側フロッピードライブ]設定の内容は考慮されません。

ユーザーがログオンしたときにフロッピードライブに自動接続できるようにするには、[クライアントドライブに自動接続する]設定を使用します。

### クライアント側ネットワークドライブ

この設定項目では、クライアント側でマップ済みのネットワークドライブ（リモートドライブ）にアクセスしたりファイルを保存したりすることを許可または禁止します。

デフォルトでは許可されます。

この設定項目をポリシーに追加するときは、[クライアントドライブリダイレクト]設定で[許可]が選択されていることを確認してください。この設定が無効な場合、クライアント側ネットワークドライブはマップされず、ユーザーが手作業でそれらのドライブにアクセスすることもできなくなります。このとき、[クライアント側ネットワークドライブ]設定の内容は考慮されません。

ユーザーがログオンしたときにネットワークドライブに自動接続できるようにするには、[クライアントドライブに自動接続する]設定を使用します。

### クライアント側光学式ドライブ

この設定項目では、クライアント側のCD-ROM、DVD-ROM、およびBD-ROMドライブにアクセスしたりファイルを保存したりすることを許可または禁止します。

デフォルトでは許可されます。

この設定項目をポリシーに追加するときは、[クライアントドライブリダイレクト]設定で[許可]が選択されていることを確認してください。この設定が無効な場合、クライアント側光学式ドライブはマップされず、ユーザーが手作業でそれらのドライブにアクセスすることもできなくなります。このとき、[クライアント側光学式ドライブ]設定の内容は考慮されません。

ユーザーがログオンしたときに光学式ドライブに自動接続できるようにするには、[クライアントドライブに自動接続する]設定を使用します。

### クライアント側リムーバブルドライブ

この設定項目により、クライアント側のUSBドライブにアクセスしたりファイルを保存したりすることを許可または禁止し

す。

デフォルトでは許可されます。

この設定項目をポリシーに追加するときは、[クライアントドライブリダイレクト]設定で[許可]が選択されていることを確認してください。この設定が無効な場合、クライアント側リムーバブルドライブはマップされず、ユーザーが手作業でこれらのドライブにアクセスすることもできなくなります。このとき、[クライアント側リムーバブルドライブ]設定の内容は考慮されません。

ユーザーがログオンしたときにリムーバブルドライブに自動接続できるようにするには、[クライアントドライブに自動接続する]設定を使用します。

### ホストからクライアントへのリダイレクト

この設定項目では、URLや特定のメディアコンテンツをクライアント側で開くためのファイルタイプの関連付けを有効または無効にします。この設定を無効にすると、コンテンツはサーバー上で開きます。

デフォルトでは無効になっています。

この設定を有効にすると、次の種類のURLがクライアント側のアプリケーションで開きます。

- Hypertext Transfer Protocol (HTTP)
- Secure Hypertext Transfer Protocol (HTTPS)
- Real PlayerおよびQuickTime (RTSP)
- Real PlayerおよびQuickTime (RTSPU)
- 従来のRealPlayer (PNM)
- Microsoft Media Server (MMS)

### クライアント側のドライブ文字を保持する

この設定項目では、クライアント側ドライブをセッション内でマップするときに、元のドライブ文字を保持するかどうかを指定します。

デフォルトでは保持されません。

この設定項目をポリシーに追加するときは、[クライアントドライブリダイレクト]設定で[許可]が選択されていることを確認してください。

### クライアント側ドライブへの読み取り専用アクセス

この設定項目では、マップされたクライアントドライブ上にユーザーやアプリケーションがファイルやフォルダーを作成したり変更したりすることを許可または禁止します。

デフォルトでは許可されます。

[有効]に設定すると、ファイルやフォルダーへの読み取り専用アクセスが許可されます。

この設定項目をポリシーに追加するときは、[クライアントドライブリダイレクト]設定で[許可]が選択されていることを確認してください。

### ユーザーフォルダーのリダイレクト

この設定項目では、Citrix ReceiverやWeb Interfaceを使用するユーザーに対して、セッション内でクライアント側の[ドキュメント]や[デスクトップ]などのローカルフォルダーに簡単にアクセスするための機能を許可または禁止します。

デフォルトでは許可されます。

この設定では、この機能の有効/無効をポリシーの適用条件に基づいて制御できます。この設定が禁止されている場合、ユーザーフォルダーのリダイレクトに関するStoreFront、Web Interface、またはCitrix Receiverのすべての設定が無視されます。

ユーザーフォルダーのリダイレクトを許可するユーザーを定義するには、この設定項目で[許可]を選択し、ポリシーの適用先としてそのユーザーを指定します。この設定は、ユーザーフォルダーのリダイレクトに関するほかの設定よりも優先されます。

ユーザーフォルダーのリダイレクトによりクライアント側のドライブがアクセスされるため、クライアント側のハードドライブへのアクセスや書き込みを禁止するとユーザーフォルダーのリダイレクトも禁止されます。

この設定項目をポリシーに追加するときは、[クライアント側固定ドライブ]設定で[許可]が選択されていることを確認してください。

### 非同期書き込みを使用する

この設定項目では、クライアント側のディスクへの非同期書き込みを有効または無効にします。

デフォルトでは無効になっています。

非同期書き込みを有効にすると、WAN接続を介したサーバーからクライアント側へのディスク書き込みおよびファイル転送の遅延が改善されます。ただし、非同期転送時にセッションが切断されたりクライアント側のディスク容量が不足したりしてファイル書き込みが中断された場合に、クライアント側のファイルが破損することがあります。この問題が発生した場合、ポップアップウィンドウが開き、影響を受けたファイルがユーザーに通知されます。ユーザーは問題を解決した後でファイル転送をやり直すことができます。

WAN接続でのファイル転送速度の改善が重要で、クライアント側のデータが破損してもユーザーが容易に復元できる場合のみ、非同期書き込みを有効にしてください。

この設定項目をポリシーに追加するときは、[クライアントドライブリダイレクト]設定で[許可]が選択されていることを確認してください。この設定が無効の場合、非同期書き込みは行われません。

# Flashリダイレクトのポリシー設定

Jan 04, 2017

[Flashリダイレクト] カテゴリには、ユーザーセッションでのFlashコンテンツの処理に関する設定項目が含まれています。

## Flashアクセラレーション

この設定項目では、Flashコンテンツのレンダリングをサーバー上ではなくクライアント側で行う機能を有効または無効にします。デフォルトでは、クライアント側でのFlashコンテンツのレンダリングが有効になっています。

注：この設定項目は、Citrix Online Plug-in 12.1でサポートされる従来のFlashリダイレクト機能について使用します。この機能を有効にすると、Flashコンテンツがユーザーデバイス上でレンダリングされるため、ネットワークおよびサーバーへの負荷が軽減されます。特定のWebサイトのFlashコンテンツを強制的にサーバー上でレンダリングするには、[Flash URL互換性リスト] 設定を使用します。

この機能を使用するには、ユーザーデバイス側でも[このユーザーデバイスでのHDX MediaStream Flashリダイレクトを有効にする] を有効にする必要があります。

Flashアクセラレーションを無効にすると、すべてのWebサイトのFlashコンテンツがサーバー上でレンダリングされます。特定のWebサイトのFlashコンテンツのみをユーザーデバイス上でレンダリングするには、[Flash URL互換性リスト] 設定を使用します。

## Flash背景色リスト

この設定項目では、特定のURLのFlashコンテンツの背景色を指定します。

デフォルトでは、背景色は指定されていません。

この背景色は、クライアント側でレンダリングされるFlashコンテンツの背景に表示され、これにより表示領域を判別しやすくなります。表示領域を判別しやすくするために、非一般的な色を指定する必要があります。

このリストには、URL（先頭および末尾にワイルドカード文字を使用可）と24ビットRGBの16進カラーコードをスペースで区切って入力します例：http://citrix.com000003。

指定されるURLはFlashコンテンツのURLである必要があります。これはWebサイトのURLとは異なることがあります。

### 警告

レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

Windows 8またはWindows 2012が動作するVDAでは、この設定によりURLのキー色の設定に失敗することがあります。この場合、VDAマシンでレジストリを編集します。

32ビットマシンの場合は、このレジストリ設定を使用します：

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\HdxMediaStreamForFlash\Server\PseudoServer]
"ForceHDXFlashEnabled"=dword:00000001
```



64ビットマシンの場合は、このレジストリ設定を使用します：

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer]
"ForceHDXFlashEnabled"=dword:00000001
```

### Flashアクセラレーションの後方互換

この設定項目は、以前のバージョンのCitrix Receiver (旧称「Citrix Online Plug-in」) 用の第1世代 (従来モード) のFlashリダイレクト機能を使用できるようにするかどうかを制御します。

デフォルトでは、有効になっています。

この機能を使用するには、ユーザーデバイス側でも [このユーザーデバイスでのHDX MediaStream Flashリダイレクトを有効にする] を有効にする必要があります。

第2世代のFlashリダイレクト機能は、Citrix Receiver 3.0で使用できます。第1世代 (従来モード) のFlashリダイレクト機能は、Citrix Online Plug-in 12.1でサポートされています。第2世代のFlashリダイレクト機能を使用するには、サーバー側およびユーザーデバイス側で第2世代のFlashリダイレクト機能を有効にする必要があります。サーバー側またはユーザーデバイス側で従来モードのFlashリダイレクト機能が有効になっている場合は、従来モードのFlashリダイレクト機能が使用されます。

### Flashアクセラレーションのデフォルトの動作

この設定では、第2世代のFlashリダイレクト機能のデフォルトの動作を設定します。

デフォルトでは、Flashアクセラレーションは有効になっています。

この設定では、次のいずれかのオプションを選択します。

- Flashアクセラレーションを有効にする：Flashリダイレクトが使用されます。
- Flash Playerをブロックする：Flashリダイレクトおよびサーバー側でのレンダリングが行われません。これにより、ユーザーはFlashコンテンツを表示できなくなります。
- アクセラレーションを無効にする：Flashリダイレクトは使用されません。適切なバージョンのFlash Player for Windows Internet Explorerがサーバーにインストールされている場合、ユーザーはサーバー側で処理されたFlashコンテンツを表示できます。

この設定は、[Flash URL互換性リスト] 設定に追加した特定のWebページやFlashインスタンスでは無視されます。また、Flashリダイレクト機能を使用するには、ユーザーデバイス側でも [このユーザーデバイスでのHDX MediaStream Flashリダイレクトを有効にする] を有効にする必要があります。

### Flashイベントのログ

この設定項目では、FlashイベントをWindowsのアプリケーションイベントログに記録するかどうかを指定します。

デフォルトでは、ログは有効になっています。

Windows 7またはWindows Vistaが動作するコンピューターでは、Flashリダイレクトのログは [アプリケーションとサービスログ] ノードに記録されます。

### Flashインテリジェントフォールバック

この設定項目は、Flashコンテンツのクライアント側でのレンダリングが不要または低速な場合に、サーバー側でのレンダリングに自動的に切り替える機能を有効または無効にします。

デフォルトでは、有効になっています。

## Flash遅延しきい値

この設定項目では、Flashコンテンツのレンダリングをサーバー側で行うかどうかを決定する遅延時間のしきい値を、0~30ミリ秒で指定します。

デフォルトのしきい値は、30ミリ秒です。

HDX MediaStream for Flashでは、起動時にサーバーとクライアント間の遅延時間が計測されます。計測値がしきい値未満の場合、HDX MediaStream for Flash機能によりクライアント側でFlashコンテンツがレンダリングされます。遅延がしきい値を超える場合は、サーバー側でFlashコンテンツがレンダリングされます（そのサーバー上にAdobe Flash Playerがインストールされている場合）。

この設定項目を有効にするときは、[Flashアクセラレーションの後方互換] 設定で [有効] が選択されていることを確認してください。

注：HDX MediaStream Flashリダイレクトを従来モードで使用する場合のみ適用されます。

## Flashビデオフォールバック防止

この設定は、「小さな」Flashコンテンツがレンダリングされてユーザーに表示されるのかどうか、されるならどのようにされるのかを指定します。

デフォルトでは、この設定は構成されていません。

この設定を構成するには、次のいずれかのオプションを選択します。

- **小さなコンテンツのみ。**サーバー上でインテリジェントフォールバックコンテンツのみレンダリングされます。ほかのFlashコンテンツはエラー\*.swfと置き換えられます。
- **サポートされたクライアントがある小さなコンテンツのみ。**クライアントが現在Flashリダイレクトを使用している場合に、サーバー上でインテリジェントフォールバックコンテンツのみレンダリングされます。ほかのFlashコンテンツはエラー\*.swfと置き換えられます。
- **サーバー側のコンテンツなし。**サーバー上のすべてのコンテンツがエラー\*.swfと置き換えられます。

このポリシー設定を使用するには、エラー\*.swfファイルを指定する必要があります。このエラー\*.swfは、VDA上でレンダリングしない任意のコンテンツを置き換えます。

## Flashビデオフォールバック防止エラー\*.swf

この設定は、サーバー負荷管理ポリシーが使用されている場合にFlashインスタンスを置き換えるためユーザーに表示されるエラーメッセージのURLを指定します。例：

```
http://domainName.tld/sample/path/error.swf
```

## Flashサーバー側でのコンテンツ取得URLリスト

この設定項目のリストに追加したWebサイトのFlashコンテンツは、サーバーにより取得され、ユーザーデバイスに転送されます。

デフォルトでは、サイトは指定されていません。

サーバー側でのコンテンツ取得は、インターネットに直接アクセスできないユーザーデバイスに対して有効にします。また、ユーザーデバイス側でも [サーバー側のコンテンツの取得を有効にする] 設定を有効にする必要があります。

第2世代のFlashリダイレクトには、FlashのSWFファイルをサーバー側でダウンロードするフォールバック機能が含まれています。ユーザーデバイス側でWebサイトからFlashコンテンツを取得できない場合は、[Flashサーバー側でのコンテンツ取得

URLリスト] 設定のURLリストに基づいて、サーバー側でコンテンツの取得が行われます。

対象のURLをリストに追加する場合、以下の点に注意してください。

- Flash Playerを開始する上位レベルのHTMLページのURLではなく、FlashコンテンツのURLを追加します。
- URLの最初または最後にワイルドカード文字としてアスタリスク (\*) を使用できます。
- URLの最後にワイルドカード文字を使うと、すべての子URLを指定できます (http://www.citrix.com/\*など)。
- 冒頭の「http://」や「https://」は正しく認識されますが、省略することもできます。

#### Flash URL互換性リスト

この設定項目では、特定のWebサイト上のFlashコンテンツをクライアント側でレンダリングするか、サーバー側でレンダリングするか、またはブロックするかを指定します。

デフォルトでは、規則は指定されていません。

対象のURLをリストに追加する場合、以下の点に注意してください。

- リスト上位のURLやアクションが優先されます。
- URLの最初または最後にワイルドカード文字としてアスタリスク (\*) を使用できます。
- URLの最後にワイルドカード文字を使うと、すべての子URLを指定できます (http://www.citrix.com/\*など)。
- 冒頭の「http://」や「https://」は正しく認識されますが、省略することもできます。
- ユーザーデバイス上で正しくレンダリングされないFlashコンテンツのサイトをこのリストに追加して、[サーバー側でレンダリング] または [ブロック] オプションを選択します。

# グラフィックのポリシー設定

Jan 04, 2017

[グラフィック] カテゴリには、ユーザーセッションでの画像処理の制御に関する設定項目が含まれています。

## 表示メモリの制限

この設定項目では、セッションのビデオバッファの最大サイズをキロバイト単位で指定します。

デフォルトでは、65536キロバイトに設定されます。

高い色数および解像度を使用するセッションでは、大きい値を指定します。必要なメモリの量は、次の式で算出できます。

必要とされるメモリ (バイト単位) = (1ピクセルあたりのビット数を8で割った色数) × (垂直方向のピクセル単位の解像度) × (水平方向のピクセル単位の解像度)

たとえば、ウィンドウの高さ600、ウィンドウの幅800、32ビットの色数の場合、必要なメモリの最大量は (32bpp÷8) × (600ピクセル) × (800ピクセル) = 1920000バイトであるため、[表示メモリの制限] 設定で1920KBを指定します。

32ビット以外の色数は、[従来のグラフィックモード] 設定が有効な場合のみ使用できます。

HDXでは、各セッションで必要な表示メモリ量だけが割り当てられます。このため、デフォルト値よりも多くのメモリが必要なユーザーが一部だけの場合にこの設定項目で表示メモリの制限を増やしても、スケーラビリティは低下しません。

## メモリが不足したときの表示モード

注: Virtual Delivery Agent 7.xでは、[従来のグラフィックモード] 設定が有効な場合のみこの設定項目が適用されます。この設定項目では、セッション表示用のメモリが上限に達したときに、色数と解像度のどちらを下げるかを指定します。

デフォルトでは、最初に色数が低下します。

セッション表示用のメモリが上限に達したときに、表示品質を下げることでメモリ不足による問題の発生を防ぐことができます。色数を下げることを選択すると、表示用のメモリが上限に達したときに、まずより少ない色でのイメージ表示に切り替わります。解像度を下げることを選択すると、まず1インチあたりのピクセル数が少なくなります。

色数または解像度の低下をユーザーに通知するには、[メモリ不足による表示品質の低下をユーザーに通知する] 設定を使用します。

## 動的ウィンドウプレビュー

この設定項目では、フリップ、フリップ3D、タスクバープレビュー、およびピークウィンドウプレビューモードにおけるシームレスウィンドウの表示の有効/無効を切り替えます。

| Windows Aeroプレビューオプション | 説明                                                             |
|------------------------|----------------------------------------------------------------|
| タスクバープレビュー             | Windowsタスクバー上のアイコン上にマウスポインターを合わせると、そのウィンドウの縮小版がプレビューとして表示されます。 |
| ピークウィンドウプレビュー          | Windowsタスクバー上に開いた縮小版上にマウスポインターを合わせると、そのウィンドウがフルサイズで表示されます。     |

|                                                        |                                                                                                                 |
|--------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <b>フリップ<br/>Windows Aeroプレ<br/>ビューオプション<br/>フリップ3D</b> | <b>説明</b><br>Alt+Tabキーを押すと、開いているすべてのウィンドウの縮小版が一覧表示されます。<br>Tab+Windowsロゴキーを押すと、開いているすべてのウィンドウが立体的に重なって一覧表示されます。 |
|--------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|

デフォルトでは、有効になっています。

### イメージキャッシュ

注：Virtual Delivery Agent 7.xでは、[従来のグラフィックモード] 設定が有効な場合のみこの設定項目が適用されます。この設定項目では、セッション内のイメージのセクションキャッシュおよび取得を有効または無効にします。必要な場合にセクションのイメージをキャッシュし、そのセクションを取得すると、スクロールがよりスムーズになり、ネットワーク上のデータ伝送量が減少して、ユーザーデバイス上で必要とされる処理が少なくなります。

デフォルトでは、イメージのキャッシュ設定は有効になっています。

注：イメージのキャッシュ設定は、イメージがどのようにキャッシュおよび取得されるかを制御します。イメージがキャッシュされるかどうかについては制御しません。従来のグラフィックモード設定が有効な場合は、イメージがキャッシュされません。

### 従来のグラフィックモード

この設定項目では、リッチなグラフィック表示を無効にして、従来の表示モードにフォールバックします。これにより、WANやモバイル接続時のスケーラビリティが向上します。

この設定項目はデフォルトで無効になっており、リッチなグラフィック表示が提供されます。

### 許可される最大表示色数

注：Virtual Delivery Agent 7.xでは、[従来のグラフィックモード] 設定が有効な場合のみこの設定項目が適用されます。この設定項目では、セッションで許可される最大表示色数を指定します。

デフォルトでは、1ピクセルあたり32ビットまでの色数が許可されます。

このフィールドはThinWireのドライバーおよび接続にのみ適用されます。Windows Display Driver Model (WDDM) ドライバーがプライマリディスプレイドライバーとして設定されているVDAなど、非ThinWireドライバーをプライマリディスプレイドライバーとして設定されているVDAには適用されません。Windows 8など、WDDMドライバーをプライマリディスプレイドライバーとして使用するデスクトップOS VDAには、この設定が影響を与えることはありません。Windows Server 2012 R2など、WDDMドライバーをプライマリディスプレイドライバーとして使用するサーバーOS VDAについては、この設定によりユーザーがVDAに接続できなくなる可能性があります。

高い表示色数をサポートするには、より多くのメモリが必要です。メモリ不足時に自動的に色数を減らすには、[メモリが不足した時の表示モード] 設定を使用します。この設定で色数を下げるオプションを選択すると、イメージの表示色数が少なくなります。

### メモリ不足による表示品質の低下をユーザーに通知する

注：Virtual Delivery Agent 7.xでは、[従来のグラフィックモード] 設定が有効な場合のみこの設定項目が適用されます。この設定項目では、色数または解像度が低下するときにユーザーに簡単なメッセージを表示するかどうかを指定します。

デフォルトでは、メッセージは表示されません。

### キューイメージの破棄

注：Virtual Delivery Agent 7.xでは、[従来のグラフィックモード] 設定が有効な場合のみこの設定項目が適用されます。この設定項目では、ほかのイメージで置換される中間イメージを破棄するかどうかを指定します。

デフォルトでは、キューイメージの破棄は有効になっています。

有効にすると、グラフィックがユーザーデバイス側に送信される際のレスポンスが向上します。ただし、中間フレームが粗落するため、アニメーションの動きがスムーズでなくなる場合があります。

# キャッシュのポリシー設定

Jan 04, 2017

[キャッシュ] カテゴリには、狭帯域幅のクライアント接続でイメージデータをユーザーデバイス上にキャッシュする機能を有効にするための設定項目が含まれています。

## 固定キャッシュしきい値

注：Virtual Delivery Agent 7.xでは、[従来のグラフィックモード] 設定が有効な場合のみこの設定項目が適用されます。この設定項目では、ビットマップをユーザーデバイスのハードドライブ上にキャッシュするときの帯域幅しきい値を指定します。この機能により、サイズの大きな、頻繁に使用されるイメージがキャッシュされ、以降のセッションで再使用されません。

デフォルトのしきい値は、3000000bpsです。

帯域幅がこのしきい値を下回る場合、固定キャッシュ機能が有効になります。つまり、デフォルトの設定では、接続帯域幅が3,000,000bpsを下回る場合に、ビットマップがユーザーデバイスのハードドライブ上にキャッシュされます。

# Keep-Aliveのポリシー設定

Jan 04, 2017

[Keep-Alive] カテゴリには、ICA Keep-Aliveメッセージの管理に関する設定項目が含まれています。

## ICA Keep-Aliveタイムアウト

この設定項目では、ICA Keep-Aliveメッセージの送信間隔を秒単位で指定します。

デフォルトでは、ICA Keep-Aliveメッセージが60秒おきに送信されます。

ICA Keep-Aliveメッセージの送信間隔として設定可能な範囲は、1~3600秒です。ただし、アイドル状態のセッションをネットワーク監視ソフトウェアで自動的に閉じるように設定している環境では、この設定項目を使用しないでください。

## ICA Keep-Alive

この設定項目では、ICA Keep-Aliveメッセージを定期的に送信するかどうかを指定します。

デフォルトでは、ICA Keep-Aliveメッセージは送信されません。

この設定を有効にすると、ネットワークの問題により切断されたセッションにユーザーが再接続できなくなることを防ぐことができます。また、サーバー側でセッションのアイドル状態が検出されたときに、リモートデスクトップサービス（RDS）によりセッションが切断されることを防ぐことができます。サーバーは、定期的にKeep-Aliveメッセージを送信して、セッションがアクティブかどうかを検出します。セッションがアクティブでないことが検出されると、サーバーにより「切断」状態として認識されます。

ICA Keep-Aliveは、セッション画面の保持機能を使用する環境では正しく動作しません。セッション画面の保持機能を使用しない環境でのみ、ICA Keep-Aliveを有効にしてください。

関連する設定項目：セッション画面の保持。



# ローカルアプリケーションアクセスのポリシー設定

Jan 04, 2017

[ローカルアプリケーションアクセス] カテゴリには、ホストされるデスクトップ環境で、ユーザーデバイス上にインストールされたローカルアプリケーションとホスト上のアプリケーションをシームレスに使用できるようにするための設定項目が含まれています。

## ローカルアプリケーションのアクセスを許可する

この設定項目では、ホストされるデスクトップ環境で、ローカルアプリケーションとホスト上のアプリケーションの統合を許可または禁止します。

ユーザーがローカルのアプリケーションを起動すると、そのアプリケーションが仮想デスクトップ上で動作しているかのように表示されます。

デフォルトでは、ローカルアプリケーションへのアクセスは禁止されます。

## URLリダイレクトのブラックリスト

この設定項目では、ユーザーデバイス上のローカルのWebブラウザで開くWebサイトを指定します。通常、ユーザーの現在位置の情報を使用するWebサイト（msn.comやnewsgoogle.comなど）や、クライアント側で処理した方が効率的なマルチメディアコンテンツサイトなどのURLを指定します。

デフォルトでは、サイトは指定されていません。

## URLリダイレクトのホワイトリスト

この設定項目では、ユーザーデバイス側にリダイレクトしないWebサイトを指定します。

デフォルトでは、サイトは指定されていません。

# モバイルデバイスでの動作のポリシー設定

Jan 04, 2017

[モバイルデバイスでの動作] カテゴリには、Citrix Mobility Packの動作を制御するための設定項目が含まれています。

## キーボードの自動表示

この設定項目では、モバイルデバイス画面上におけるキーボードの自動表示を有効または無効にします。

デフォルトでは、無効になっています。

## タッチパネルでの操作に最適化されたデスクトップ

この設定はデフォルトで無効になっており、Windows 10マシンでは使用できません。

この設定では、タッチパネルでの操作に最適化されたデスクトップの起動を許可または禁止して、全体的なReceiverインターフェイスの動作を指定します。

デフォルトでは、タッチパネルでの操作に最適化されたデスクトップが起動します。

通常のWindowsインターフェイスのデスクトップを起動する場合は、[禁止] を選択します。

## コンボボックスをデバイス側で表示する

この設定項目では、モバイルデバイスでのセッションで表示するコンボボックスの種類を指定します。モバイルデバイス側のコンボボックスコントロールを表示するには、[許可] を選択します。管理者がこの設定項目で許可を選択しても、Receiver for iOSのユーザーは、セッション設定で通常のWindowsコンボボックスの表示を選択できます。

デフォルトでは、コンボボックスをデバイス側で表示する機能は禁止されています。

# マルチメディアのポリシー設定

Jan 04, 2017

[マルチメディア] カテゴリには、ユーザーセッションでのオーディオとビデオのストリーム配信の管理に関する設定項目が含まれています。

## ビデオ品質の制限

この設定項目では、HDX接続で許可される最大ビデオ品質レベルを指定します。最大ビデオ品質を指定すると、マルチメディアコンテンツに対する一定レベルのQoS (Quality of Service) を保証できます。

デフォルトでは、未構成になっています。

許可される最大ビデオ品質レベルを指定するには、次のいずれかのオプションを選択します。

- 1080p/8.5mbps
- 720p/4.0mbps
- 480p/720kbps
- 380p/400kbps
- 240p/200kbps

注：単一サーバー上で複数のビデオを同時に再生すると多くのリソースが消費され、サーバーのスケーラビリティが低下することがあります。

## マルチメディア会議

この設定項目では、ビデオ会議アプリケーションの使用を許可または禁止します。

デフォルトでは、許可されます。

この設定項目をポリシーに追加するときは、[Windows Mediaリダイレクト] 設定で [許可] が選択されていることを確認してください。

マルチメディア会議を使用する場合、次の条件を満たしていることを確認してください。

- マルチメディア会議に使用するWebカメラの製造元が提供するドライバーがインストール済みである。
- ビデオ会議セッションを始める前に、Webカメラがユーザーデバイスに接続済みである。複数のWebカメラを同時に使用することはできません。ユーザーデバイス上に複数のWebカメラが装着されている場合、最初に検出されたものから接続が試行されます。

## WAN接続でのWindows Mediaマルチメディアリダイレクトの最適化

この設定項目を許可するとリアルタイムマルチメディアトランスコードが有効になり、オーディオやビデオのメディアコンテンツをモバイルデバイスにストリーム配信したり、WAN上でのWindows Mediaコンテンツの配信を最適化してユーザーエクスペリエンスを向上させたりします。

デフォルトでは、WANを介したWindows Mediaコンテンツの配信が最適化されます。

この設定項目をポリシーに追加するときは、[Windows Mediaリダイレクト] 設定で [許可] が選択されていることを確認してください。

この設定項目を有効にすると、メディアのストリーム配信を有効にするリアルタイムマルチメディアトランスコードが必要に応じて自動的に適用され、ネットワーク条件が悪い場合でもシームレスなユーザーエクスペリエンスが提供されます。

## WAN接続でのWindows MediaマルチメディアリダイレクトでのGPUの使用

この設定項目では、リアルタイムマルチメディアトランスコードの処理をVirtual Delivery Agent (VDA) 側のGPU (Graphics Processing Unit) で行い、サーバーのスケラビリティを向上させる機能を許可または禁止します。GPUでのトランスコード処理は、VDA側にハードウェアアクセラレーションをサポートするGPUが搭載されている場合にのみ可能になります。適切なGPUがない場合は、CPUがトランスコード処理を行います。

注：GPUでのトランスコード処理は、NVIDIA社のGPUでのみサポートされます。

デフォルトでは、WANを介したWindows Mediaコンテンツ配信をVDA側のGPUを使用して最適化する機能は禁止されています。

この設定項目をポリシーに追加するときは、[Windows Mediaリダイレクト] 設定および [WAN接続でのWindows Mediaマルチメディアリダイレクトの最適化] 設定で [許可] が選択されていることを確認してください。

## ビデオフォールバック防止

管理者はビデオフォールバック防止ポリシー設定を使って、ストリーム配信されたコンテンツをユーザーに配信する方法を設定できます。

デフォルトでは、この設定は構成されていません。これにより、サーバー側でのフォールバックへのRAVEに対するクライアント側での取得を可能にします。

この設定を構成するには、次のいずれかのオプションを選択します。

- **Server Fetched - Server Rendered**。サーバー側でのフォールバックへのRAVEに対するクライアント側での取得を可能にします。
- **Server Fetched - Client Rendered**。RAVEフォールバックに対するクライアント側での取得を可能にしますが、サーバー側でのレンダリングフォールバックへのRAVEをブロックします。
- **Client Fetched - Client Rendered**。サーバー側でのレンダリングフォールバックへのRAVEに対するクライアント側での取得をブロックします。

コンテンツを再生しない場合、プレーヤーのウィンドウにエラーメッセージ「Company has blocked video because of lack of resources」が表示されます。

## Windows Mediaのクライアント側でのコンテンツ取得

この設定項目では、インターネットやイントラネット上のマルチメディアコンテンツなどをホストサーバーを介さず直接ユーザーデバイス上にストリーム配信する機能を許可または禁止します。

デフォルトでは、許可されます。

この設定項目を許可すると、マルチメディアファイルがホストサーバーではなくユーザーデバイス側で処理されるため、ネットワーク消費が効率化され、サーバースケラビリティが向上します。また、ユーザーデバイス上にMicrosoft DirectShowやMedia Foundationなどの高度なマルチメディアフレームワークをインストールする必要もなくなり、マルチメディアをURLから再生する機能だけが使用されます。

この設定項目をポリシーに追加するときは、[Windows Mediaリダイレクト] 設定で [許可] が選択されていることを確認してください。[Windows Mediaリダイレクト] 設定で [禁止] を選択すると、Windows Mediaのクライアント側でのコンテンツ取得機能も無効になります。

## Windowsメディアリダイレクト

この設定項目では、サーバーでのストリーミングオーディオとビデオの配信方法を制御および最適化します。

デフォルトでは、ストリーミングオーディオとビデオのユーザーデバイスへのストリーム配信が許可されます。

この設定項目を許可すると、セッション内で再生されるオーディオおよびビデオの品質が向上して、ユーザーデバイス上のファイルを再生しているときの品質に近くなります。マルチメディアデータはサーバーからユーザーデバイスに、元の圧縮されたままの形で配信され、ユーザーデバイス側でメディアの展開およびレンダリングが行われます。

Windows Media Directでは、Microsoft社のDirectShow、DirectX Media Objects (DMO)、およびMedia Foundation規格に準拠するコーデックでエンコードされたマルチメディアファイルが最適化されます。ユーザーデバイス側でメディアファイルの展開およびレンダリングを行うため、そのファイルのエンコーディング形式をサポートするコーデックがユーザーデバイス上にインストールされている必要があります。

デフォルトでは、Citrix Receiverでオーディオサポートは無効になっています。ユーザーがICAセッション内でマルチメディアアプリケーションを実行できるようにするには、管理者がオーディオのサポートを有効にして、ユーザーがReceiverのオーディオ機能を有効にする必要があります。

Windows Media Directによるメディアの再生品質が、基本的なICA圧縮および通常のオーディオ機能での品質よりも悪い場合は、[禁止]を選択します。キーフレームの周波数が著しく低いメディアデータを狭帯域幅接続で再生する場合などで、この機能による問題がまれに生じることがあります。

#### Windows Media Directのバッファサイズ

この設定項目では、マルチメディアアクセラレーションのバッファサイズを1~10秒の間で指定します。

デフォルトのバッファサイズは5秒です。

#### Windows Media Directのバッファサイズ使用

この設定項目では、[Windows Media Directのバッファサイズ]設定で指定したバッファサイズを有効または無効にします。

デフォルトでは、指定されているデフォルトバッファサイズが使用されません。

この設定項目で[無効]を選択した場合、または[Windows Media Directのバッファサイズ]設定が未構成の場合、デフォルトのバッファサイズ(5秒)が適用されます。

# マルチストリーム接続のポリシー設定

Jan 04, 2017

[マルチストリーム接続] カテゴリには、単一セッションでの複数ICA接続のQoS (Quality of Service) 優先度の管理に関する設定項目が含まれます。

## UDPを使用したオーディオ

この設定項目では、サーバーのUDPを使用したオーディオを許可または禁止します。

デフォルトでは許可されます。

この機能を有効にすると、サーバー上のUDPポートが開き、[UDPでのオーディオリアルタイムトランスポート] 設定が有効なすべての接続でそのポートが使用されます。

## オーディオUDPポートの範囲

この設定項目では、Virtual Delivery Agent (VDA) でユーザーデバイスとのオーディオパケットデータの送受信に使用されるポート番号の範囲を「.」形式で指定します。VDAでは、オーディオデータの送受信に各UDPポートペアの使用が試行されます。まず最小のポート番号が使用され、以降の試行では2ずつ番号を増やしていきます。各ポートは、受信トラフィックと送信トラフィックの両方に使用されます。

デフォルトでは、「16500,16509」に設定されています。

## マルチポートポリシー

この設定項目では、ICAトラフィックで使用されるTCPポートおよび各ポートのネットワーク優先度を指定します。

デフォルトでは、プライマリポート (2598) に優先度 [高] が設定されています。

ポートには、以下の優先度を設定できます。

- 最高 - Webカメラを使ったビデオ会議など、リアルタイムプロセスに適しています。
- 高 - 画面、キーボード、マウスなど、インタラクティブなトラフィックに適しています。
- 中 - クライアント側ドライブのマッピング機能など、バルクプロセスに適しています。
- 低 - 印刷など、バックグラウンドプロセスに適しています。

各ポートには異なる優先度を設定する必要があります。つまり、CGPポート1とCGPポート3の両方で優先度に [最高] を設定することはできません。

ポートの優先度設定を削除するには、ポート番号として「0」を入力します。プライマリポートの優先度設定を削除したり変更したりすることはできません。

この設定項目をポリシーに追加したら、サーバーを再起動します。この設定項目は、コンピューターポリシーの[マルチストリーム] 設定が有効な場合のみ適用されます。

## マルチストリームコンピューター設定

この設定項目では、サーバーのマルチストリーム機能を有効または無効にします。

デフォルトでは、無効になっています。

Citrix CloudBridgeでマルチストリーム機能をサポートする場合は、この設定項目を使用する必要はありません。この設定項

目は、サードパーティ製のルーターや従来のBranch Repeaterを使用する環境でQoS（Quality of Service）優先度を指定するときに使用できます。

この設定項目の変更を反映させるには、サーバーを再起動する必要があります。

**重要：**この設定項目を、帯域幅を制限する設定項目（[セッション全体の最大帯域幅] など）と一緒に使用すると、意図したとおりに動作しなくなる場合があります。ポリシーでこの設定項目を使用する場合は、帯域幅を制限する設定項目を構成しないでください。

#### マルチストリームユーザー設定

この設定項目では、ユーザーデバイスのマルチストリーム機能を有効または無効にします。

デフォルトでは、すべてのユーザーに対して無効になっています。

この設定項目は、[マルチストリームコンピューター設定]設定が有効なホストに対してのみ適用されます。

**重要：**この設定項目を、帯域幅を制限する設定項目（[セッション全体の最大帯域幅] など）と一緒に使用すると、意図したとおりに動作しなくなる場合があります。ポリシーでこの設定項目を使用する場合は、帯域幅を制限する設定項目を構成しないでください。

# ポートリダイレクトのポリシー設定

Jan 04, 2017

[ポートリダイレクト] カテゴリには、クライアント側のLPTポートおよびCOMポートのマッピングに関する設定項目が含まれています。

注：Virtual Delivery Agent 7.xでは、レジストリを使用してこれらの機能を構成します（「[レジストリを使ったCOMポートおよびLPTポートリダイレクト設定の構成](#)」参照）。

## クライアントCOMポートを自動接続する

この設定項目では、ユーザーのログオン時にクライアント側のCOMポートに自動的に接続する機能を有効または無効にします。

デフォルトでは無効になっています。

## クライアントLPTポートを自動接続する

この設定項目では、ユーザーのログオン時にクライアント側のLPTポートに自動的に接続する機能を有効または無効にします。

デフォルトでは無効になっています。

## クライアントCOMポートリダイレクト

この設定項目では、COMポートのクライアント側へのリダイレクトを許可または禁止します。

デフォルトでは禁止されます。

関連する設定項目は以下のとおりです。

- COMポートリダイレクトの最大帯域幅 (Kbps)
- COMポートリダイレクトの最大帯域幅 (%)

## クライアントLPTポートリダイレクト

この設定項目では、LPTポートのクライアント側へのリダイレクトを許可または禁止します。

デフォルトでは禁止されます。

LPTポートは、印刷ジョブをユーザーデバイス上のプリンターオブジェクトではなくLPTポートに送信するレガシーアプリケーションでのみ使用されます。最近のアプリケーションでは、LPTポートではなくプリンターオブジェクトに印刷ジョブが送信されます。この設定項目は、LPTポートへの出力を行うレガシーアプリケーションをホストするサーバーに対してのみ使用します。

関連する設定項目は以下のとおりです。

- LPTポートリダイレクトの最大帯域幅 (Kbps)
- LPTポートリダイレクトの最大帯域幅 (%)



# 印刷のポリシー設定

Jan 04, 2017

[印刷] カテゴリには、クライアントからの印刷の管理に関する設定項目が含まれています。

## クライアントプリンターリダイレクト

この設定項目では、ユーザーのログオン時にクライアントプリンターをサーバーに自動的にマップすることを許可または禁止します。

デフォルトでは許可されます。この設定項目が無効の場合、PDFプリンターはセッションで自動作成されません。

関連する設定項目：クライアントプリンターを自動作成する

## デフォルトプリンター

この設定項目では、セッションのデフォルトのクライアントプリンターとして設定するプリンターを指定します。

デフォルトでは、ユーザーの現在のデフォルトプリンター（通常使うプリンター）がセッションのデフォルトプリンターとして使用されます。

[デフォルトプリンターの設定を変更しない] を選択すると、リモートデスクトップサービスまたはWindowsのユーザープロファイルで設定されているデフォルトプリンターが使用されます。この場合、デフォルトプリンターはプロファイルに保存されず、ほかのセッションやクライアント側のプロパティにより変更されなくなります。このオプションでは、セッションで最初に自動作成されたプリンターがセッションのデフォルトプリンターになります。つまり、以下のどちらかのプリンターになります。

- Windowsサーバーのコントロールパネルの [デバイスとプリンター] でローカルに追加された最初のプリンター。
- サーバーにローカルプリンターが追加されていない場合は、最初に自動作成されたプリンター。

プロファイルの設定に基づいてユーザーに最も近いプリンターを提供する（近接プリンター機能を使用する）場合に、このオプションを使用できます。

## プリンター割り当て

この設定項目は、[デフォルトプリンター] 設定および [セッションプリンター] 設定の代わりに使用します。特定のサイト、大規模グループ、または組織単位用のポリシーを構成する場合は、[デフォルトプリンター] 設定および [セッションプリンター] 設定を使用します。[プリンター割り当て] 設定は、多くのプリンターのグループを複数のユーザーに割り当てる場合に使用します。

この設定項目では、ユーザーデバイスを一覧に追加して、そのユーザーデバイス上のデフォルトプリンターがセッションでどのように使用されるかを指定します。

デフォルトでは、ユーザーの現在のデフォルトプリンター（通常使うプリンター）がセッションのデフォルトプリンターとして使用されます。

また、各ユーザーデバイスに対してセッションで自動作成するネットワークプリンターを指定します。デフォルトでは、プリンターは指定されていません。

- デフォルトプリンター値は、以下のように設定します。  
ユーザーデバイスの現在のデフォルトプリンターを使用する場合は、[変更しない] を選択します。

現在のリモートデスクトップサービスまたはWindowsのユーザープロファイルで設定されているデフォルトプリンターを

使用する場合は、[変更しない]を選択します。この場合、デフォルトプリンターはプロファイルに保存されず、ほかのセッションやクライアント側のプロパティにより変更されなくなります。このオプションでは、セッションで最初に自動作成されたプリンターがセッションのデフォルトプリンターになります。つまり、以下のどちらかのプリンターになります。

- Windowsサーバーのコントロールパネルの[デバイスとプリンター]でローカルに追加された最初のプリンター。
- サーバーにローカルプリンターが追加されていない場合は、最初に自動作成されたプリンター。
- セッションプリンター値を設定するには、自動作成するプリンターのUNCパスを入力します。この一覧の設定は、ユーザーがログオンするたびに適用できます。

### プリンター自動作成イベントログの設定

この設定項目では、プリンターの自動作成処理中にログに記録するイベントを指定します。エラーおよび警告をログに記録しない、エラーのみを記録する、またはエラーおよび警告を記録することを選択できます。

デフォルトでは、エラーおよび警告がログに記録されます。

たとえば、プリンターのネイティブドライバーをインストールできず、代わりにユニバーサルプリンタードライバーがインストールされた場合は、警告がログに記録されます。このような状況でユニバーサルプリンタードライバーを使用できるようにするには、[ユニバーサル印刷の使用]設定で[ユニバーサル印刷のみを使用する]または[要求されたドライバーを使用できない場合にのみユニバーサル印刷を使用する]を選択します。

### セッションプリンター

この設定項目では、セッションで自動作成するネットワークプリンターを指定します。

デフォルトでは、プリンターは指定されていません。

自動作成するプリンターを追加するには、そのプリンターのUNCパスを入力します。この一覧の設定は、ユーザーがログオンするたびに適用できます。

### プリンターの自動作成を待機する (サーバーデスクトップ)

この設定では、サーバーデスクトッププリンターが自動作成されるまでセッションへの接続を遅延させるかどうかを指定します。

デフォルトでは、プリンターの作成を待機せずに接続します。

# クライアントプリンターのポリシー設定

Jan 04, 2017

[クライアントプリンター] カテゴリには、クライアントプリンターに関する設定項目が含まれています。これには、クライアントプリンターの自動作成、プリンタープロパティの保存、およびプリントサーバーへの接続のための設定が含まれています。

## クライアントプリンターを自動作成する

この設定では、自動作成するクライアントプリンターを指定します。この設定は、デフォルトのクライアントプリンター自動作成設定より優先されます。

デフォルトでは、すべてのクライアントプリンターが自動作成されます。

この設定は、[クライアントプリンターリダイレクト]設定で[許可]が選択されている場合にのみ適用されます。

この設定では、次のオプションを選択します。

- [すべてのクライアントプリンターを自動作成する]では、ユーザーデバイス上のすべてのプリンターが自動作成されます。
- [デフォルトのクライアントプリンターのみを自動作成する]では、ユーザーデバイス上のデフォルトプリンターのみが自動作成されます。
- [ローカル（ネットワークを介さない）クライアントプリンターのみを自動作成する]では、ユーザーデバイスのローカルポート（LPTポート、COMポート、USBポート、TCP/IPポートなど）に直接接続されているプリンターのみが自動作成されます。
- [クライアントプリンターを自動作成しない]では、ユーザーがログオンするときのすべてのクライアントプリンターの自動作成が無効になります。リモートデスクトップサービス（RDS）で設定されているクライアントプリンターの自動作成オプションが適用されるようにするには、このオプションを選択して、そのポリシーの優先度をほかのポリシーよりも高くします。

## 汎用ユニバーサルプリンターを自動作成する

注：このポリシー設定問題を解決するためのHotfixがCitrix Knowledge CenterのCTX141565およびCTX141566で提供されています。

この設定項目では、UDPをサポートするクライアントのセッションでCitrixユニバーサルプリンターの汎用印刷オブジェクトを自動作成する機能を有効または無効にします。

デフォルトでは、汎用ユニバーサルプリンターオブジェクトは自動作成されません。

関連する設定項目は以下のとおりです。

- ユニバーサル印刷の使用
- ユニバーサルドライバーの優先度

## クライアントプリンター名

この設定では、自動作成されるクライアントプリンターの命名規則を選択します。

デフォルトでは、標準のプリンター名が使用されます。

[標準のプリンター名]を選択して、「HPLaserjet 4 fromclientnamein session 3」などのプリンター名を使用します。

MetaFrame Presentation Server 3.0またはそれ以前のバージョンと互換性のある命名規則でクライアントプリンターを作成す

るには、[従来のプリンター名]を選択します。この場合、「Client/clientname#/HPLaserjet 4」などの名前が使用されま  
す。ただし、セキュリティ上の理由から、やむを得ない場合を除きこのオプションを使用しないことをお勧めします。

注：このオプションは、以前のバージョンのXenAppやXenDesktopとの互換性を保持する場合に使用します。

#### プリントサーバーへの直接接続

この設定項目では、ネットワーク共有上のクライアントプリンターを使用するときに、クライアントを経由せずに仮想デスク  
トップやホストサーバーからプリントサーバーに直接接続することを有効または無効にします。

デフォルトでは有効になっています。

仮想デスクトップやホストサーバーとネットワークプリントサーバーが同一LAN上にあり、WANで隔たれていない場合に直  
接接続を有効にします。この場合、仮想デスクトップやホストサーバーからLANを介してプリントサーバーに直接印刷データ  
転送されるため、処理が高速になります。

仮想デスクトップやホストサーバーとネットワークプリントサーバーがWANで隔たれていたり、遅延や帯域幅の問題が生じ  
たりする場合は、直接接続を無効にできます。直接接続を無効にすると、印刷ジョブがユーザーデバイスに送信され、そこ  
からネットワークプリントサーバーにリダイレクトされます。ユーザーデバイスに送信されるデータは圧縮されるため、デー  
タがWANを横断するときに消費される帯域幅が少なくなります。

同じ名前を持つネットワークプリンターが2つ存在する場合は、ユーザーデバイスと同じネットワーク上のプリンターが使用  
されます。

#### プリンタードライバーのマッピングと互換性

この設定項目では、自動作成されるクライアントプリンターのドライバー置換規則を指定します。

デフォルトでは、規則は指定されていません。

ドライバー置換規則を定義すると、プリンターの自動作成時に特定のドライバーの使用を許可したり、ユニバーサル印刷の使  
用を指定したりできます。ドライバーの置換規則では、サーバーとクライアント間でドライバー名をマップして、ユーザー  
デバイスから提供されるプリンタードライバーではなくサーバー上のドライバーが使用されるように設定します。これにより、  
サーバー側のドライバーとクライアント側のドライバーの名前が異なっても、サーバー上のアプリケーションからクラ  
アントプリンターに出力できるようになります。

ドライバー置換規則の一覧では、ドライバーマッピングの追加、既存のマッピングの編集、マッピングに対するカスタム設定  
の上書き、マッピングの削除、および一覧のドライバーエントリの順序の変更を実行できます。マッピングを追加するには、  
クライアント側プリンタードライバーの名前を入力し、それを置換するサーバー側プリンタードライバーを選択します。

#### プリンタープロパティの保存

この設定項目では、ユーザーが変更したクライアントプリンターのプロパティをどこに保存するかを指定します。

デフォルトでは、システムの判定により、クライアントデバイスに保存できない場合にのみユーザープロファイルにプリン  
タープロパティが保存されます。

この設定項目では、次のオプションを選択します。

- [クライアントデバイスにのみ保存する]は、更新されないユーザープロファイル（固定プロファイルや移動プロファイル）  
を使用する環境で選択します。このオプションは、サーバーファーム内のすべてのサーバーでXenApp 5以降が動作し  
ており、ユーザーがCitrix Online Plug-in Versions 9~12.x、またはCitrix Receiver 3.xを使用する場合にのみ選択してくださ  
い。
- [ユーザープロファイルにのみ保存する]は、使用帯域幅とログオン速度に制限があるユーザーデバイス（このオプショ  
ンではネットワークトラフィックが軽減されます）、または古いプラグインソフトウェアを使用するユーザーのためのオ

プションです。このオプションでは、サーバー上のユーザープロファイルにプリンタープロパティを保存し、ユーザーデバイス上のプロパティを使用しません。このオプションは、Presentation Server 3.0またはそれ以前のバージョンと、Presentation Serverクライアント8.x以前が使用される環境で選択してください。ただし、このオプションはリモートデスクトップサービス（RDS）の移動プロファイルにのみ適用されます。

- [クライアントに保存できない場合にのみユーザープロファイルに保存する]では、システムによりプリンタープロパティの保存先が決定されます。ユーザーデバイスに保存できない場合にのみ、ユーザープロファイルにプリンタープロパティが保存されます。さまざまな環境やクライアントの条件に対応できるオプションですが、システムチェック処理が行われるため、ログオン時に遅延が生じたり使用帯域幅が増えたりすることがあります。
- [プリンタープロパティを保持しない]を選択した場合、プリンタープロパティは保持されません。

## クライアントプリンターの保持と復元

この設定項目では、ユーザーデバイス上のプリンターをセッション間で保持および再作成する機能を有効または無効にします。デフォルトでは、クライアントプリンターは自動的に保持および復元されます。

「保持されるプリンター」とは、ユーザーが作成し次回セッションの開始時に再作成されるプリンターを指します。保持されるプリンターがXenAppにより再作成されるときは、[クライアントプリンターを自動作成する]設定以外のすべてのポリシー設定が考慮されます。

「復元されるプリンター」とは、管理者がカスタマイズしクライアントポートに永続的に接続された状態で保存されるプリンターを指します。

# ドライバーのポリシー設定

Jan 04, 2017

[ドライバー] カテゴリには、プリンタードライバーに関する設定項目が含まれています。

## 付属のプリンタードライバーの自動インストール

この設定項目では、Windowsに付属のドライバーセットや、pnputil.exe /aによりホスト上にステージングされたドライバーパッケージから、プリンタードライバーを必要に応じて自動的にインストールする機能を有効または無効にします。

デフォルトでは、自動インストールが有効になっています。

## ユニバーサルドライバーの優先度

この設定項目では、ユニバーサルプリンタードライバーの使用優先順位を指定します。一覧の上位にあるドライバーから順使用されます。

デフォルトの優先順位は以下のとおりです。

- EMF
- XPS
- PCL5c
- PCL4
- PS

この一覧では、ドライバーを追加、編集、または削除したり、優先順位を変更したりできます。

## ユニバーサル印刷の使用

この設定項目では、どのような状況でユニバーサル印刷を使用するかを指定します。

デフォルトでは、要求されたドライバーを使用できない場合にのみユニバーサル印刷を使用します。

ユニバーサル印刷では、プリンター固有の標準ドライバーの代わりに汎用プリンタードライバーが使用されるため、ホスト：コンピューターでのドライバー管理がシンプルになります。ユニバーサルプリンタードライバーを使用できるかどうかは、ユーザーデバイス、ホスト、およびプリントサーバーソフトウェアにより決定されます。構成によっては、ユニバーサル印刷を使用できない場合があります。

この設定項目では、次のオプションを選択します。

- [プリンター固有のドライバーのみを使用する]では、クライアントプリンターの自動作成時に、そのプリンター固有の標準プリンタードライバーが使用されます。必要なプリンタードライバーがサーバーにない場合、そのクライアントプリンターは自動作成されません。
- [ユニバーサル印刷のみを使用する]を有効にすると、プリンター固有の標準ドライバーは使用されません。ユニバーサルプリンタードライバーのみを使用してプリンターが作成されます。
- [要求されたドライバーを使用できない場合にのみユニバーサル印刷を使用する]では、可能な場合はプリンター固有の標準ドライバーが使用されます。プリンター固有のドライバーがサーバーにない場合は、最適なユニバーサルドライバーを使用してクライアントプリンターが自動作成されます。
- [ユニバーサル印刷を使用できない場合にのみプリンター固有のドライバーを使用する]では、可能な場合はユニバーサルプリンタードライバーが使用されます。ユニバーサルプリンタードライバーがサーバーにない場合は、適切なプリンター固有の標準ドライバーを使用してクライアントプリンターが自動作成されます。

# Universal Print Serverのポリシー設定

Jan 04, 2017

[Universal Print Server] カテゴリには、Universal Print Serverの動作を制御するための設定項目が含まれています。

## Universal Print Serverの有効化

この設定項目では、仮想デスクトップや公開アプリケーションでのUniversal Print Server機能を有効または無効にします。この設定項目を仮想デスクトップまたはアプリケーションのホストサーバーを含んでいる組織単位 (OU) に割り当てます。

デフォルトでは、Universal Print Serverは無効になっています。

この設定項目では、以下のいずれかのオプションを選択します。

- **有効。Windowsのリモート印刷機能にフォールバックする**：ネットワークプリンターへの接続時にUniversal Print Serverが使用されます。Universal Print Serverを使用できない場合は、Windowsの印刷プロバイダーが使用されます。Windows印刷プロバイダーにより作成されたすべてのプリンターは、引き続きWindows印刷プロバイダーによって処理されます。
- **有効。Windowsのリモート印刷機能にフォールバックしない**：ネットワークプリンターへの接続時にUniversal Print Serverのみが使用されます。Universal Print Serverを使用できない場合は、ネットワークプリンターの接続に失敗します。この設定により、Windowsの印刷プロバイダーを使用したネットワーク印刷を禁止できます。Windows印刷プロバイダーにより作成されたプリンターは、この設定項目が構成されたポリシーがアクティブな間は作成されなくなります。
- **無効**。Universal Print Server機能が無効になります。UNC名のネットワークプリンターに接続するときに、Universal Print Serverによる接続は試行されません。リモートプリンターへの接続では、Windowsのリモート印刷機能が引き続き使用されます。

## Universal Print Server印刷データストリーム (CGP) ポート

この設定項目では、Universal Print Server印刷データストリームCGP (Common Gateway Protocol) リスナーが使用するTCPポート番号を指定します。この設定項目を構成したポリシーは、プリントサーバーを含んでいる組織単位に割り当てます。

デフォルトのポート番号は、7229に設定されています。

ほかのポートを指定する場合は、1から65535の番号を使用してください。

## Universal Print Server入力データストリームの最大帯域幅 (Kbps)

この設定項目では、各印刷ジョブからUniversal Print ServerにCGPで配信される印刷データの転送速度の上限をキロビット/秒単位で指定します。この設定項目を構成したポリシーは、仮想デスクトップまたはアプリケーションのホストサーバーを含んでいる組織単位に割り当てます。

デフォルトでは、上限なし (0) が指定されています。

## Universal Print Server Webサービス (HTTP/SOAP) ポート

この設定では、Universal Print Server Webサービス (HTTP/SOAP) リスナーで使用されるTCPポート番号を指定します。Universal Print Serverはオプションコンポーネントで、ネットワークプリンターでのCitrixユニバーサルプリントドライバーの使用を有効にします。Universal Print Serverが使用されると、印刷コマンドがSOAP over HTTP上のSOAPを経由してXenAppおよびXenDesktopホストからUniversal Print Serverに送信されます。この設定は、Universal Print ServerがHTTP/SOAP要求を受信するためリスンするデフォルトのTCPポートを変更します。

ホストおよびプリントサーバーのHTTPポートの両方を等しく構成する必要があります。ポートを同じように構成しないと、ホストソフトウェアがUniversal Print Serverに接続しません。この設定は、XenAppおよびXenDesktop上のVDAを変更しま

す。また、Universal Print Serverのデフォルトのポートを変更する必要があります。

デフォルトのポート番号は、8080に設定されています。

ほかのポートを指定する場合は、0から65535の番号を使用してください。



# ユニバーサル印刷のポリシー設定

Jan 04, 2017

[ユニバーサル印刷] カテゴリには、ユニバーサル印刷の管理に関する設定項目が含まれています。

## ユニバーサル印刷EMF処理モード

この設定項目では、Windowsユーザーデバイス上でのEMFスプールファイルの処理方法を制御します。

デフォルトでは、EMFスプールファイルがクライアント上のスプールキューに直接挿入されます。

この設定項目では、次のオプションを選択します。

- [EMFスプールファイルを再処理する] を有効にすると、EMFスプールファイルが再処理され、ユーザーデバイス上のGDIサブシステム経由で送信されます。通常、EMF再処理を必要とするドライバーは自動的に検出され、適切な印刷経路が使用されますが、セッションで正しく検出されない場合があります。そのような場合にこのオプションを選択します。
- Citrixユニバーサルプリンタードライバーで[EMFスプールファイルを直接挿入する] を有効にすると、EMFレコードがホスト上でスプールされ、そのEMFスプールファイルがユーザーデバイス側に送信され処理されます。通常、このEMFスプールファイルはクライアント上のスプールキューに直接挿入されます。EMF形式を処理できるプリンターおよびドライバーでは、この方法により印刷を高速に実行できます。

## ユニバーサル印刷イメージ圧縮制限

この設定項目では、Citrixユニバーサルプリンタードライバーでのイメージ印刷で使用できる品質レベルの上限を指定します。

デフォルトでは、イメージ品質の上限が[最高品質（無損失圧縮）] に設定されています。

[非圧縮] を選択すると、EMF印刷では圧縮が無効になります。

この設定項目では、次のオプションを選択します。

- 非圧縮
- 最高品質（無損失圧縮）
- 高品質
- 標準品質
- 低品質（最大圧縮）

この設定項目を [ユニバーサル印刷最適化デフォルト] と同じポリシーに追加する場合は、次の点に注意してください。

- [ユニバーサル印刷イメージ圧縮制限] での圧縮レベルが [ユニバーサル印刷最適化デフォルト] での設定よりも低い場合は、[ユニバーサル印刷イメージ圧縮制限] の圧縮レベルが適用されます。
- [ユニバーサル印刷イメージ圧縮制限] で [非圧縮] を選択すると、[ユニバーサル印刷最適化デフォルト] の [必要なイメージ品質] および [ヘビーウェイト圧縮を有効にする] オプションの設定は無視されます。

## ユニバーサル印刷最適化デフォルト

この設定項目では、セッションで作成されるユニバーサルプリンタードライバーのデフォルトの印刷最適化オプションを指定します。

- [必要なイメージ品質] では、ユニバーサル印刷に適用されるイメージ圧縮レベルの上限を指定します。デフォルトでは [標準品質] が選択されており、ユーザーは標準品質または低品質（最大圧縮）を使ってイメージを印刷できます。
- [ヘビーウェイト圧縮を有効にする] では、ヘビーウェイト圧縮を有効または無効にします。この機能では、画質を損なわずに [必要なイメージ品質] での圧縮レベルよりも高い帯域幅削減が提供されます。デフォルトでは、ヘビーウェイト圧縮は無効になっています。

- [イメージおよびフォントのキャッシュ]では、印刷ストリームで使用されているイメージやフォントをキャッシュするかどうかを指定します。キャッシュを有効にすると、同一のイメージやフォントがプリンターに複数回送信されることを防ぐことができます。デフォルトでは、埋め込みイメージおよびフォントがキャッシュされます。これらの設定は、ユーザーデバイスでその機能をサポートしている場合にのみ適用されます。
- [非管理者によるこれらの設定の変更を許可する]では、非管理者ユーザーがセッション内でこれらの最適化設定を変更することを許可または禁止します。デフォルトでは、禁止されています。

注：これらのすべてのオプションは、EMF印刷に対してのみ適用されます。XPS印刷では、[必要なイメージ品質] オプションのみがサポートされます。

この設定項目を [ユニバーサル印刷イメージ圧縮制限] と同じポリシーに追加する場合は、次の点に注意してください。

- [ユニバーサル印刷イメージ圧縮制限] での圧縮レベルが [ユニバーサル印刷最適化デフォルト] での設定よりも低い場合は、[ユニバーサル印刷イメージ圧縮制限] の圧縮レベルが適用されます。
- [ユニバーサル印刷イメージ圧縮制限] で [非圧縮] を選択すると、[ユニバーサル印刷最適化デフォルト] の [必要なイメージ品質] および [ヘビーウェイト圧縮を有効にする] オプションの設定は無視されます。

### ユニバーサル印刷プレビューの設定

この設定項目では、自動作成プリンターまたは汎用ユニバーサルプリンターの印刷プレビュー機能を使用するかどうかを指します。

デフォルトでは、自動作成プリンターまたは汎用ユニバーサルプリンターの印刷プレビューは使用できません。

この設定項目では、次のオプションを選択します。

- 自動作成プリンターまたは汎用ユニバーサルプリンターの印刷プレビューを使用しない
- 自動作成プリンターの印刷プレビューのみを使用する
- 汎用ユニバーサルプリンターの印刷プレビューのみを使用する
- 自動作成プリンターおよび汎用ユニバーサルプリンターの印刷プレビューを使用する

### ユニバーサル印刷品質制限

この設定項目では、セッションでの印刷出力で使用できる最大DPI値（インチあたりのドット数）を指定します。

デフォルトでは [制限なし] が選択されており、ユーザーは接続しているプリンターで許可されている最高印刷品質を選択できます。

そのほかの値を選択すると、ユーザーが使用できる出力解像度が制限されます。この設定項目では、印刷品質自体と、ユーザーが接続するプリンターの印刷能力の両方が制限されます。たとえば、[中解像度 (600dpi)] を選択した場合、ユーザーの印刷出力の最高品質は600DPIに制限され、ユニバーサルプリンターの [詳細設定] タブの印刷品質設定には、中品質 (600DPI) を超える解像度オプションが表示されなくなります。

この設定項目では、次のオプションを選択します。

- ドラフト (150dpi)
- 低解像度 (300dpi)
- 中解像度 (600dpi)
- 高解像度 (1200dpi)
- 制限なし

# セキュリティのポリシー設定

Jan 04, 2017

[セキュリティ] カテゴリには、セッションの暗号化とログオンデータの暗号化の構成に関する設定項目が含まれていません。

## SecureICAの最低暗号化レベル

この設定項目では、サーバーとユーザーデバイス間で送信するセッションデータの暗号化に必要な最低限の暗号化レベルを指定します。

### 重要：

Virtual Delivery Agent 7.xの場合、この設定項目をRC5 128ビット暗号化によるログオンデータの暗号化を有効にするためだけに使用できます。ほかの暗号化レベルは、以前のバージョンのXenAppやXenDesktopとの互換性を保持する場合に使用しません。

VDA 7.xの場合、セッションデータの暗号化はVDAのデリバリーグループの基本設定を使って設定されます。デリバリーグループに対して [Secure ICAを有効にする] がオンになっている場合、セッションデータはRC5 (128ビット) 暗号化で暗号化されます。デリバリーグループに対して [Secure ICAを有効にする] がオフになっている場合、セッションデータは基本レベルの暗号化で暗号化されます。

この設定項目では、次のオプションを選択します。

- [基本] では、非RC5のアルゴリズムを使ってクライアント接続を暗号化します。この暗号化レベルでは、データストリームが直接読み取られることはありませんが、解読される恐れがあります。デフォルトでは、クライアントとサーバー間のトラフィックには基本レベルの暗号化が使用されます。
- [RC5 (128ビット、ログオンのみ)] では、RC5 128ビット暗号化を使ってログオンデータを暗号化し、基本レベルの暗号化を使ってクライアント接続を暗号化します。
- [RC5 (40ビット)] では、RC5 40ビット暗号化を使ってクライアント接続を暗号化します。
- [RC5 (56ビット)] では、RC5 56ビット暗号化を使ってクライアント接続を暗号化します。
- [RC5 (128ビット)] では、RC5 128ビット暗号化を使ってクライアント接続を暗号化します。

クライアントとサーバー間の実際の通信では、Citrix製品やWindowsオペレーティングシステムでの暗号化設定も考慮されます。サーバーやユーザーデバイスでより高い暗号化レベルが設定されている場合は、その設定が優先されます。

機密データを使用するユーザーなど、特定のユーザーの通信データを保護してメッセージの整合性を保証するために、より高度な暗号化レベルを設定することもできます。ポリシーでより高度な暗号化レベルを指定すると、そのレベルよりも低い暗号化機能を使用するReceiverは、サーバーに接続できなくなります。

SecureICAでは認証の実行またはデータの整合性のチェックはされません。エンドツーエンドの暗号化を提供するには、SecureICAをSSLまたはTLSと共に使用します。

SecureICAではFIPS準拠のアルゴリズムは使用されません。このことが問題になる場合は、SecureICAを使用しないようにサーバーとReceiverを設定してください。

# サーバーの制限のポリシー設定

Jan 04, 2017

[サーバーの制限] カテゴリには、アイドル状態の接続の制御に関する設定項目が含まれています。

## サーバーのアイドルタイマーの間隔

この設定項目では、アイドル状態のセッション（ユーザーからの入力がない連続セッション）を自動的に切断するまでの時間をミリ秒単位で指定します。

デフォルトでは、アイドル状態の接続は切断されません。つまり、サーバーのアイドルタイマーの間隔は0です。

### 注意

このポリシー設定が使用される場合、セッションが指定した時間アイドル状態になると、「アイドルタイマーが切れました」ということを示すダイアログボックスがユーザーに表示されることがあります。これはCitrixポリシー設定によって制御されるものではなく、Microsoftのダイアログボックスです。詳しくは、<http://support.citrix.com/article/CTX118618>を参照してください。

# セッションの制限のポリシー設定

Jan 04, 2017

[セッションの制限] カテゴリには、セッションに接続してから強制的にログオフさせられるまでの時間を制御するための設定項目が含まれています。

## 切断セッションタイマー

この設定項目では、切断状態でロックされたデスクトップセッションを一定期間後に自動的にログオフする機能を有効または無効にします。

デフォルトでは、切断状態のセッションはログオフされません。

## 切断セッションタイマーの間隔

この設定項目では、切断状態でロックされたデスクトップセッションを自動的にログオフするまでの時間を分単位で指定します。

デフォルトでは、1440分（24時間）に設定されています。

## セッション接続タイマー

この設定項目では、ユーザーデバイスとデスクトップ間の連続セッションを一定期間後に自動的にログオフする機能を有効または無効にします。

デフォルトでは、無効になっています。

## セッション接続タイマーの間隔

この設定項目では、ユーザーデバイスとデスクトップ間の連続セッションを自動的にログオフするまでの時間を分単位で指定します。

デフォルトでは、1440分（24時間）に設定されています。

## セッションアイドルタイマー

この設定項目では、ユーザーからの入力がないユーザーデバイスとデスクトップ間の連続セッションを一定期間後に自動的にログオフする機能を有効または無効にします。

デフォルトでは、有効になっています。

## セッションアイドルタイマーの間隔

この設定項目では、ユーザーからの入力がないユーザーデバイスとデスクトップ間の連続セッションを自動的にログオフするまでの時間を分単位で指定します。

デフォルトでは、1440分（24時間）に設定されています。

# セッション画面の保持のポリシー設定

Jan 04, 2017

[セッション画面の保持] カテゴリには、セッション画面の保持の管理に関する設定項目が含まれています。

## セッション画面の保持

この設定項目では、セッション画面の保持機能を許可または禁止します。

デフォルトでは、許可されます。

セッション画面の保持機能は、ICAセッションをアクティブのまま保持し、ネットワークの接続が切断されても、セッションの画面を表示したままにできます。ユーザーは、接続が回復するまでセッション画面を見ることができます。

セッション画面の保持機能を有効にすると、データを損失することなく、サーバー上のセッションがアクティブのまま保持されます。ネットワークが中断されると、セッション画面が停止し、カーソルの形が砂時計に変わるため、ユーザーもネットワークが切断されていることがわかります。このとき、セッションウィンドウが閉じたりエラーメッセージが表示されたりする代わりに画面表示が保持され、バックグラウンドで再接続が試行されます。ネットワーク接続が回復すると、自動的にセッションでの作業を再開できるようになります。また、セッションに再接続するときに再認証用のログオン画面が表示されないため、ユーザーは即座に作業を再開できます。切断したセッションに再接続するユーザーを再認証する場合は、[クライアントの自動再接続時の認証] 設定を使用します。これにより、ユーザーが再接続するときに資格情報を入力するための画面が表示されます。

セッション画面の保持機能とクライアントの自動再接続機能を一緒に使用する場合は、次のように処理されます。まず、ネットワークが切断されると、セッション画面の保持機能により、セッションがアクティブのままサーバー上に保持されます。[セッション画面の保持のタイムアウト] 設定で指定した時間が経過すると、サーバー上のセッションが終了または切断されます。この後でクライアントの自動再接続のポリシー設定により、切断セッションへの再接続が行われます。

## セッション画面の保持のポート番号

この設定項目では、セッション画面の保持機能で使用する、受信TCPポートを指定します。

デフォルトのポート番号は、2598に設定されています。

## セッション画面の保持のタイムアウト

この設定項目では、セッション画面の保持機能でセッションをアクティブのままサーバー上に保持する時間を秒単位で指定します。ここで指定した時間が経過しても再接続されないセッションは、「切断セッション」として処理されます。

デフォルトでは、180秒 (3分) に設定されています。

この時間を長く設定することもできますが、この機能の本来の目的は、ネットワークから切断されたユーザーを再認証することなくセッションに再接続することにあるので注意が必要です。必要以上に長い時間を設定すると、接続の再開を待ちきれないユーザーが席を離れてしまい、その間に不正なユーザーがセッションにアクセスしてしまう危険性があります。

# タイムゾーン制御のポリシー設定

Jan 04, 2017

[タイムゾーン制御] カテゴリには、セッションでのローカルタイムの使用に関する設定項目が含まれています。

## レガシークライアントのローカルタイムゾーンを検出する

この設定項目では、クライアント側のローカルタイムゾーンの検出を有効または無効にします。クライアントによっては、正確なタイムゾーン情報がサーバーに送信されない場合があります。

デフォルトでは、必要に応じてクライアント側のタイムゾーンが検出されます。

この設定は、詳細なタイムゾーン情報をサーバーに送信しない古いReceiverまたはICAクライアントと共に使用することを目的としています。サポートされるバージョンのReceiver for Windowsなど、詳細なタイムゾーン情報をサーバーに送信するReceiverと共に使用しても、この設定には効果がありません。

## クライアントのローカルタイムゾーンを使用する

この設定項目では、ユーザーセッションに適用されるタイムゾーンを指定します。ユーザーセッションにサーバー側のタイムゾーンを適用したり、ユーザーデバイス側のタイムゾーンを適用したりできます。

デフォルトでは、ユーザーセッションのタイムゾーンが適用されます。

この機能を使用するには、グループポリシーエディターの[タイムゾーンリダイレクトを許可する]設定（[ユーザーの構成] > [管理用テンプレート] > [Windowsコンポーネント] > [リモートデスクトップサービス] > [リモートデスクトップセッションホスト] > [デバイスとリソースのリダイレクト]）を有効にしてください。

# TWAINデバイスのポリシー設定

Jan 04, 2017

[TWAINデバイス] カテゴリには、デジタルカメラやスキャナーなどのクライアントTWAINデバイスのマッピングと、サーバーからクライアントへのイメージ転送の最適化に関する設定項目が含まれています。

## 注意

現在、TWAIN 2.0 はサポートされていません。

### クライアントTWAINデバイスリダイレクト

この設定項目では、サーバー上でホストされるアプリケーションから、クライアント側に接続されているデジタルカメラなどのTWAINデバイスにアクセスすることを許可または禁止します。デフォルトでは許可されます。

関連する設定項目は以下のとおりです。

- TWAIN圧縮レベル
- TWAINデバイスリダイレクトの最大帯域幅 (Kbps)
- TWAINデバイスリダイレクトの最大帯域幅 (%)

### TWAIN圧縮レベル

この設定項目では、クライアントからサーバーに転送される画像の圧縮レベルを指定します。画質を最高にするには [低] を、良好にするには [中] を、低くするには [高] を選択します。デフォルトでは、中レベルの圧縮が選択されています。



# USBデバイスのポリシー設定

Jan 04, 2017

[USBデバイス] カテゴリには、USBデバイスのファイルリダイレクトの管理に関する設定項目が含まれています。

## クライアントUSBデバイス最適化規則

XenAppおよびXenDesktop 7.6 FP3で、クライアントUSBデバイス最適化規則をデバイスに適用して最適化を無効にしたり、最適化モードを変更したりできます。

ユーザーがUSB入力デバイスを接続すると、デバイスがUSBポリシー設定で許可されているかどうかホストがチェックします。デバイスが許可されている場合は、次にホストはデバイスのクライアント**USBデバイス最適化規則**をチェックします。規則が指定されていない場合は、デバイスはインタラクティブモード (02) で制御されます。キャプチャモード (04) は署名デバイスに対する推奨モードです。使用できるモードについては、以下の説明を参照してください。

## ヒント

- Wacom署名パッドおよびタブレットの使用する場合、スクリーンセーバーを無効にすることをお勧めします。その実行方法については、このセクションの最後で説明します。
- WacomSTU署名パッドおよびタブレット製品シリーズの最適化のサポートは、XenAppおよびXenDesktop 7.6 FP3のXenAppおよびXenDesktopポリシーのインストールで事前構成されています。
- 署名デバイスはXenAppおよびXenDesktopで動作し、署名デバイスとして使用するためのドライバーは必要ありません。Wacomには、デバイスをよりカスタマイズするためインストールできる追加のソフトウェアがありません。<http://www.wacom.com/>を参照してください。
- 描画用タブレット。ある種の描画入力デバイスはPCI/ACPIバス上のHIDデバイスとされ、サポートされていません。これらのデバイスについては、クライアント上のUSBホストコントローラーで接続し、XenDesktopセッション内でリダイレクトする必要があります。

ポリシー規則は、スペースで区切ったtag=value式の形式にします。以下のタグがサポートされます。

| タグ名 | 説明                                                                                                                                                                                                      |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| モード | 最適化モードは、class= <b>03</b> の入力デバイスに対してサポートされません。サポートされるモード：<br><br>最適化なし - 値 <b>01</b> 。<br><br>対話モード - 値 <b>02</b> 。ペンタブレットや3D Proマウスなどのデバイスにお勧めします。<br><br>キャプチャモード - 値 <b>04</b> 。署名パッドなどのデバイスに推奨しません。 |
| VID | デバイス記述子のベンダーID                                                                                                                                                                                          |
| PID | デバイス記述子の製品ID                                                                                                                                                                                            |
| REL | デバイス記述子のリリースID                                                                                                                                                                                          |

|          |                                |
|----------|--------------------------------|
| Class    | Classデバイス記述子またはインターフェイス記述子のクラス |
| SubClass | デバイス記述子またはインターフェイス記述子のサブクラス    |
| Prot     | デバイス記述子またはインターフェイス記述子のプロトコル    |

## 例

Mode=00000004 VID=1230 PID=1230 class=03 #Input device operating in capture mode

Mode=00000002 VID=1230 PID=1230 class=03 #Input device operating in interactive mode (default)

Mode=00000001 VID=1230 PID=1230 class=03 #Input device operating without any optimization

Mode=00000100 VID=1230 PID=1230 # Device setup optimization disabled (default)

Mode=00000200 VID=1230 PID=1230 # Device setup optimization enabled

## レジストリ設定を使った最適化モードの無効化

レジストリフラグを使って、システム全体で最適化モードを無効にできます。

HKLM\System\CurrentControlSet\Services\Icausb\Parameters

**DisableInputOptimization** DWORD - 値を**1**に設定します

このレジストリの変更を適用するため、システムの再起動が必要です。

## Wacom署名パッドデバイスのスクリーンセーバーの無効化

Wacom署名パッドおよびタブレットを使用する場合、次の手順に従ってスクリーンセーバーを無効にすることをお勧めします。

1. デバイスのリダイレクト後に**Wacom-STU-Driver**をインストールします。
2. **Wacom-STU-Display MSI**をインストールして、署名パッドコントロールパネルへのアクセスを有効にします。
3. **[Control Panel]**、**[Wacom STU Display]**、**[STU430]** または **[STU530]** の順に選択し、使用しているモデルのタブを選択します。
4. UACセキュリティウィンドウがポップアップ表示されたら **[Change]** をクリックしてから **[Yes]** をクリックします。
5. **[Disable slideshow]** を選択して、**[Apply]** をクリックします。

1つの署名パッドモデルに対しての設定が完了したら、それがすべてのモデルに適用されます。

## クライアントUSBデバイスリダイレクト

この設定項目では、USBデバイスのクライアント側へのリダイレクトおよびクライアント側からのリダイレクトを許可または禁止します。

デフォルトでは、USBデバイスはリダイレクトされません。

## クライアントUSBデバイスリダイレクト規則

この設定では、USBデバイスのリダイレクト規則を指定します。

デフォルトでは、規則は指定されていません。

ユーザーがUSBデバイスを装着すると、ホストデバイスで一覧の規則が順に検証され、マッチする最初の規則でリダイレクトが許可されているかどうかチェックされます。最初の一致がAllow規則の場合、そのUSBデバイスが仮想デスクトップで使用可能になります。最初の一致がDeny規則の場合、そのUSBデバイスはローカルデスクトップでのみ使用可能になります。一致する規則がない場合、デフォルトの規則が使用されます。

ポリシー規則は、{Allow:|Deny:}の後に、「タグ=値」式をスペースで区切って設定します。以下のタグがサポートされません。

| タグ名      | 説明                             |
|----------|--------------------------------|
| VID      | デバイス記述子のベンダーID                 |
| PID      | デバイス記述子の製品ID                   |
| REL      | デバイス記述子のリリースID                 |
| Class    | Classデバイス記述子またはインターフェイス記述子のクラス |
| SubClass | デバイス記述子またはインターフェイス記述子のサブクラス    |
| Prot     | デバイス記述子またはインターフェイス記述子のプロトコル    |

新しいポリシー規則を作成する場合、以下の点に注意してください。

- 大文字と小文字は区別されません。
- 規則の末尾に、#で始まる任意のコメントを追加できます。
- 空白行およびコメントのみの行は無視されます。
- タグには等号(=)を使用する必要があります(「VID=1230」など)。
- 各規則を1行ずつ記述するか、同一行に記述する場合はセミコロンで区切られたリスト形式である必要があります。
- USBクラスコードについては、USB Implementers Forum, Inc.のWebサイトを参照してください。

管理者定義のUSBポリシー規則の例

- ALLOW: VID=1230 PID=0007 # ANOther Industries, ANOther Flash Drive
- DENY: Class=08 subclass=05 # Mass Storage
- すべてのUSBデバイスのリダイレクトを禁止する場合は、タグを指定せずに「DENY:」を使用します。

クライアントUSBプラグアンドプレイデバイスリダイレクト

この設定項目では、カメラやPOS (Point-Of-Sale) デバイスなど、プラグアンドプレイデバイスのセッション内での使用を許可または禁止します。

デフォルトでは、許可されます。[許可]を選択すると、特定のユーザーやグループのセッションですべてのプラグアンドプレイデバイスがリダイレクトされます。[禁止]を選択すると、デバイスはリダイレクトされません。

# 視覚表示のポリシー設定

Jan 04, 2017

「視覚表示」カテゴリには、仮想デスクトップからユーザーデバイスに送信されるイメージの品質を制御するための設定項目が含まれています。

## シンプルなグラフィック用の優先色数

XenAppおよびXenDesktop 7.6 FP3にのみ適用されます。

シンプルなグラフィックではピクセルあたり16ビットに設定して色数を低下させ、画質のわずかに犠牲にすることで低帯域幅接続での応答性を潜在的に向上させることができます。このオプションは、グラフィックの圧縮にビデオコーデックが使用されていない場合にのみサポートされます。

デフォルトでは、これはピクセルあたり24ビットに設定されています。

## ターゲットフレーム数

この設定項目では、仮想デスクトップからユーザーデバイスに送信されるイメージの1秒あたりの最大フレーム数 (fps) を設定します。

デフォルトの最大フレーム数は、30fpsです。

1秒あたりのフレーム数を高く (30など) すると、ユーザーエクスペリエンスは向上しますが、より多くの帯域幅が必要になります。1秒あたりのフレーム数を低く (10など) すると、ユーザーエクスペリエンスは低下しますが、サーバーのスケールビリティが向上します。CPUが低速なユーザーデバイスに対しては、小さい値を指定した方がユーザーエクスペリエンスが向上する場合があります。

## 圧縮にビデオコーデックを使用

XenAppおよびXenDesktop 7.6 FP3にのみ適用されます。

エンドポイントでビデオのデコードを使用できる場合は、グラフィックの圧縮にビデオコーデックを使用できます。エンドポイントでビデオのデコードを使用できない、またはビデオコーデックを使用しないように指定すると、静止画像圧縮とビットマップキャッシュの組み合わせが使用されます。

デフォルトでは、使用できる場合にはビデオコーデックを使用するように設定されています。

## 表示品質

この設定では、ユーザーデバイス側に表示されるイメージの表示品質を指定します。

デフォルトでは、[中] に設定されています。

イメージの表示品質を指定するには、次のいずれかのオプションを選択します。

- **低**
- **中** - 一般的に最良のパフォーマンスおよび帯域幅効率が提供されます。
- **高** - 視覚的に無損失なイメージ品質が提供されます。
- **操作時は低品質** - 多くのネットワークトラフィックが発生している間は非可逆イメージが送信され、ネットワークトラフィックが減少したときに高品質な無損失イメージが送信されます。この設定により、帯域幅を制限されたネットワーク接続でのパフォーマンスが向上します。
- **常に無損失** - X線写真を表示するなど、イメージの画質が優先される場合では、[常に無損失] を選択します。これによ

り、イメージデータの損失を避けることができます。

[従来のグラフィックモード] 設定で [有効] を指定したポリシーに [表示品質] 設定を構成しても無視されます。

# 動画のポリシー設定

Jan 04, 2017

[動画] カテゴリには、動画の圧縮機能を無効にしたり変更したりするための設定項目が含まれています。

## 画質の下限レベル

注：Virtual Delivery Agent 7.xでは、[従来のグラフィックモード] 設定が有効な場合のみこの設定項目が適用されます。この設定項目では、アダプティブ表示の最低レベルの画質を指定します。圧縮率が小さいほど、画質は高くなります。超最高、最高、高、通常、または低から選択します。

デフォルトでは、[通常] に設定されています。

## 動画圧縮

この設定では、アダプティブ表示を有効または無効にします。アダプティブ表示機能では、ビデオやスライドショーのスライド切り替え時の画質が、使用可能な帯域幅に基づいて自動的に調節します。アダプティブ表示を有効にすると、表示品質を劣化させることなくプレゼンテーションをスムーズに実行できます。

デフォルトでは、アダプティブ表示が有効になっています。

VDAのバージョン7.0~7.6では、従来のグラフィックモードが有効な場合のみこの設定が適用されます。VDAのバージョン7.6 FP1、FP2、およびFP3については、従来のグラフィックモードが有効の場合、あるいは従来のグラフィックモードが無効でグラフィックの圧縮にビデオコーデックが使用されていない場合、この設定が適用されます。

従来のグラフィックモードが有効な場合、ポリシーの変更を適用する前にセッションを再起動する必要があります。アダプティブ表示とプログレッシブ表示は相互に排他的です。アダプティブ表示を有効にすると、プログレッシブ表示は無効になり、その逆の場合も同じです。ただし、プログレッシブ表示とアダプティブ表示の両方を同時に無効にすることは可能です。従来からの機能であるプログレッシブ表示はXenAppまたはXenDesktopにはお勧めしません。プログレッシブしきい値レベルを設定するとアダプティブ表示は無効になります。

## プログレッシブ圧縮のレベル

注：Virtual Delivery Agent 7.xでは、[従来のグラフィックモード] 設定が有効な場合のみこの設定項目が適用されます。この設定項目では、最初にダウンロードされるイメージの品質を落として、初期表示のパフォーマンスを向上させる機能を制御します。

デフォルトでは、プログレッシブ圧縮は適用されません。

プログレッシブ圧縮では、初期表示の後で、より詳細なイメージデータがダウンロードされます（そのイメージの圧縮レベルは非可逆圧縮設定で制御されます）。[最高] または [超最高] を選択すると、写真など帯域幅に負荷のかかるグラフィックの表示パフォーマンスが向上します。

プログレッシブ圧縮による効果を得るには、[非可逆圧縮のレベル] よりも高い圧縮レベルを指定する必要があります。

注：プログレッシブ表示の圧縮レベルを高くすると、セッションでの動的イメージの対話操作性が向上します。この機能を有効にすると、3Dモデルを回転させる場合など、イメージを動かしている間の表示品質は一時的に低下します。イメージを停止させると、非可逆圧縮のレベルで制御される画質が適用されます。

関連する設定項目は以下のとおりです。

- プログレッシブ圧縮のしきい値
- プログレッシブヘビーウェイト圧縮

## プログレッシブ圧縮のしきい値

注：Virtual Delivery Agent 7.xでは、[従来のグラフィックモード] 設定が有効な場合のみこの設定項目が適用されます。この設定項目では、プログレッシブ圧縮を適用する接続の最大帯域幅を、キロビット/秒 (Kbps) 単位で指定します。この帯域幅に達しないクライアント接続にのみ圧縮が適用されます。

デフォルトのしきい値は、2147483647KB/秒です。

関連する設定項目は以下のとおりです。

- プログレッシブ圧縮のしきい値
- プログレッシブヘビーウェイト圧縮

#### 保持する最低フレーム数

この設定では、低帯域幅接続時に確保される動的イメージの最低フレーム数を、フレーム数/秒 (fps) 単位で指定します。デフォルトでは、10fpsに設定されています。

VDA 7.0~7.6では、[従来のグラフィックモード] が有効な場合のみこの設定が適用されます。VDA 7.6 FP1~FP3では、[従来のグラフィックモード] が有効な場合のみこの設定が適用されます。

# 静止画のポリシー設定

Jan 04, 2017

[静止画] カテゴリには、静止画の圧縮機能を無効にしたり変更したりするための設定項目が含まれています。

## エクストラ色圧縮

この設定では、狭帯域幅接続でのイメージ配信で使用されるエクストラ色圧縮機能を有効または無効にします。この機能を有効にすると、イメージ品質が低下しますが狭帯域幅接続におけるセッションの応答性が向上します。デフォルトでは、エクストラ色圧縮は無効になっています。

エクストラ色圧縮を有効にした場合、[エクストラ色圧縮しきい値] の設定値を下回るクライアント接続でのみこの圧縮機能が適用されます。クライアント接続の帯域幅がしきい値を上回る場合、または [エクストラ色圧縮] 設定で [無効] が選択されている場合、この圧縮機能は適用されません。

## エクストラ色圧縮しきい値

注 : Virtual Delivery Agent 7.xでは、[従来のグラフィックモード] 設定が有効な場合のみこの設定項目が適用されます。この設定項目では、エクストラ色圧縮を適用する接続の最大帯域幅を、キロビット/秒 (Kbps) 単位で指定します。クライアント接続帯域幅がこの値を下回る場合、エクストラ色圧縮が適用されます ( [エクストラ色圧縮] 設定で [有効] が選択されている場合) 。

デフォルトのしきい値は、8192KB/秒です。

## ヘビーウェイト圧縮

注 : Virtual Delivery Agent 7.xでは、[従来のグラフィックモード] 設定が有効な場合のみこの設定項目が適用されます。この設定項目では、プログレッシブ圧縮よりもさらに消費帯域幅を節約する機能を有効または無効にします。ヘビーウェイト圧縮では、CPU要求度の高いグラフィックアルゴリズムが使用され、画質を損なわずにイメージデータで使用される帯域を抑えることができます。

デフォルトでは、ヘビーウェイト圧縮は無効になっています。

この圧縮機能を有効にすると、すべての非可逆圧縮設定に適用されます。この機能はCitrix Receiverでサポートされ、ほかのプラグインソフトウェアでは無視されます。

関連する設定項目は以下のとおりです。

- プログレッシブ圧縮のレベル
- プログレッシブ圧縮のしきい値

## 非可逆圧縮のレベル

注 : Virtual Delivery Agent 7.xでは、[従来のグラフィックモード] 設定が有効な場合のみこの設定項目が適用されます。この設定項目では、狭帯域幅接続でのイメージ配信で使用される非可逆圧縮のレベルを指定します。狭帯域幅接続では、ICAセッション内での非圧縮イメージの表示に時間がかかる場合があります。

デフォルトでは、中レベルの圧縮が選択されています。

イメージ表示のパフォーマンスを改善させるには、高い圧縮レベルを使用します。逆に、X線写真を表示するなどイメージの画質が優先される場合では、非可逆圧縮を無効にします。

関連する設定項目 : 非可逆圧縮のしきい値



## 非可逆圧縮のしきい値

注：Virtual Delivery Agent 7.xでは、[従来のグラフィックモード] 設定が有効な場合のみこの設定項目が適用されます。この設定項目では、非可逆圧縮を適用する接続の最大帯域幅を、キロビット/秒 (Kbps) 単位で指定します。

デフォルトのしきい値は、2147483647KB/秒です。

非可逆圧縮のしきい値を指定せずに [非可逆圧縮のレベル] 設定をポリシーに追加すると、LAN環境での高精細ビットマップ (写真など) の表示速度が向上する場合があります。

関連する設定項目：非可逆圧縮のレベル

# WebSocketのポリシー設定

Jan 04, 2017

[WebSocket] カテゴリには、Receiver for HTML5を使用した仮想デスクトップおよびホストアプリケーションへのアクセスに関する設定項目が含まれています。WebSocket機能により、Webブラウザアプリケーションとサーバー間の双方向通信が有効になります。複数のHTTP接続を確立する必要がないため、セキュリティが向上し、サーバーのオーバーヘッドが軽減されます。

## WebSocket接続

この設定項目では、WebSocketプロトコルによる接続を許可または禁止します。

デフォルトでは、無効になっています。

## WebSocketポート番号

この設定項目では、WebSocket接続の着信ポートの番号を指定します。

デフォルトのポート番号は、8080に設定されています。

## WebSocket信頼される接続元サーバー一覧

この設定項目では、信頼される接続元サーバー（通常Receiver for Web）のURLをコンマ区切りの一覧で指定します。この一覧に追加したサーバーからのWebSocket接続のみが受け入れられます。

デフォルトでは、ワイルドカード文字「\*」が設定されています。これにより、すべてのReceiver for WebサイトURLが信頼され、アクセスが許可されます。

この設定項目では、URLを以下の形式で指定します。

<protocol>://<hostFQDN>[:port]

ここで、はHTTPまたはHTTPSです。にポート番号を指定しない場合、HTTPでは80、HTTPSでは443が使用されます。

URLの一部にワイルドカード文字「\*」を使用できますが、IPアドレスの一部には使用できません（「10.105.\*.\*」は無効です）。

# 負荷管理のポリシー設定

Jan 04, 2017

[負荷管理] カテゴリには、Windowsサーバーマシン間の負荷を管理するための設定項目が含まれています。

## 同時ログオントレランス

この設定項目では、サーバーが許容できる同時ログオンの最大数を指定します。

デフォルトでは、「2」に設定されています。

## CPU使用率

この設定項目では、サーバーを「負荷限界」とみなすCPU使用率をパーセンテージで指定します。この設定項目を有効にすると、デフォルトで90%になったときにそのサーバーが負荷限界として認識されます。

デフォルトでは無効になっており、CPU使用率が負荷計算から除外されます。

## CPU使用率から除外するプロセスの優先順位

この設定項目では、特定の優先度レベル以下のプロセスをCPU使用率の負荷計算から除外できます。

デフォルトでは、[通常以下および低]に設定されています。

## ディスク使用率

この設定項目では、サーバーを「75%の負荷状態」とみなすディスクキューの長さを指定します。この設定項目を有効にすると、デフォルトでディスクキューの長さが8になったときにそのサーバーの負荷が75%であると認識されます。

デフォルトでは無効になっており、ディスク使用率が負荷計算から除外されます。

## 最大セッション数

この設定項目では、サーバーがホストできる最大セッション数を指定します。この設定項目を有効にすると、デフォルトで最大250個のセッションを単一サーバーでホストできます。

デフォルトでは、有効になっています。

## メモリ使用率

この設定項目では、サーバーを「負荷限界」とみなすメモリ使用率をパーセンテージで指定します。この設定項目を有効にすると、デフォルトで90%になったときにそのサーバーが負荷限界として認識されます。

デフォルトでは無効になっており、メモリ使用率が負荷計算から除外されます。

## 基本メモリ使用量

この設定項目では、オペレーティングシステムの基本メモリ使用量をMB単位で指定します。この値を下回ると、サーバーに負荷なしとみなされます。

デフォルトでは、768 MBに設定されています。

# プロファイル管理のポリシー設定

Jan 04, 2017

〔プロファイル管理〕カテゴリには、プロファイル管理機能を有効にして、その処理の対象として特定のグループを追加したり除外したりするための設定項目が含まれています。

これらの設定項目に対応するINIファイルの名前や、各設定項目をサポートするProfile managementのバージョン要件などの情報については、「[Profile managementポリシー](#)」を参照してください。

# 上級設定のポリシー設定

Jan 04, 2017

[上級設定] カテゴリには、Profile managementの詳細構成に関する設定項目が含まれています。

## 自動構成を無効にする

この設定項目では、Profile managementによるグループポリシーの自動構成機能を無効にします。Profile managementは、環境の構成（Personal vDiskが存在するかどうかなど）を検査してそれに基づいてグループポリシーを自動的に構成します。この機能では、未構成のProfile management関連の設定項目だけが自動構成され、カスタマイズした既存の設定は保持されます。これにより、短時間での展開と容易な最適化が可能になります。この自動構成機能には特別な構成は必要ありません。アップグレード（既存の設定を保持する場合）やトラブルシューティングを行うときは、自動構成機能を無効にすることができます。この自動構成機能は、XenAppやほかの環境では使用できません。

デフォルトでは、自動構成が許可されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、自動構成機能によりProfile management関連の設定が変更されることがあります（環境の構成に変更があった場合）。

## 問題が発生する場合にユーザーをログオフ

この設定項目では、Profile managementでユーザーストアを使用できないなどの問題が発生したユーザーを自動的にログオフする機能を有効または無効にします。この設定項目を有効にすると、プロファイルに関する問題が発生したユーザーにエラーメッセージが表示され、強制的にログオフされます。この設定項目を無効にすると、そのようなユーザーには一時プロファイルが提供されます。

この設定項目はデフォルトで無効になっており、問題が発生したユーザーに一時プロファイルが提供されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、一時プロファイルが提供されます。

## ロックされたファイルにアクセスする場合の試行数

この設定項目では、ロックされたファイルにProfile managementがアクセスするときの試行数を指定します。

デフォルトでは、5回に設定されています。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、デフォルト値が使用されます。

## ログオフ時のインターネットCookieファイルの処理

この設定項目では、ログオフ時にProfile managementでindex.datを処理して、ファイルシステムに残っているインターネットCookieを削除する機能を有効または無効にします。これにより、継続的なWebブラウズによるCookieでプロファイルが膨張することを避けることができます。ただし、この処理によりログオフに時間がかかることがあるため、問題が生じた場合のみこの設定項目を有効にしてください。

この設定項目はデフォルトで無効になっており、Profile managementはログオフ時にindex.datを処理しません。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、index.datは処理されません。

# 基本設定のポリシー設定

Jan 04, 2017

[基本設定] カテゴリには、Profile managementの基本構成に関する設定項目が含まれています。

## アクティブライトバック

この設定項目では、更新されたファイルやフォルダー（レジストリ設定は除く）をセッション中、つまりユーザーがログオフする前にユーザーストアに同期する機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、この機能が有効になります。

## Profile managementの有効化

この設定項目では、Profile managementによるログオンおよびログオフのプロファイル処理を有効または無効にします。

デフォルトでは、展開を容易にするために無効になっています。

**重要：** Profile managementを有効にする前に、ほかのすべてのセットアップタスクを実行し、Citrixユーザープロファイルの動作をテストすることをお勧めします。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、WindowsユーザープロファイルはProfile managementで処理されません。

## 除外グループ

この設定項目では、Profile managementでのプロファイル処理から除外するコンピューターのローカルグループおよびドメイングループ（ローカル、グローバル、およびユニバーサル）を指定します。

この設定項目を有効にすると、指定したユーザーグループのプロファイルがProfile managementで処理されなくなります。

この設定項目はデフォルトで無効になっており、すべてのユーザーグループのプロファイルが処理されます。

ドメイングループは、「<DOMAIN NAME>\<GROUP NAME>」形式で指定します。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、すべてのユーザーグループのプロファイルが処理されます。

## オフラインプロファイルサポート

この設定項目では、ネットワークから切断したときのプロファイル管理に関するオフラインプロファイルサポート機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目は、移動するデバイス（ラップトップやモバイルデバイス）のユーザーに適しています。ネットワークの切断が発生した場合、再起動や休止状態後もプロファイルがラップトップコンピューターやモバイルデバイス上にそのまま保持されます。ネットワークが切断されたままユーザーが作業する場合、プロファイルはローカルで更新され、ネットワーク接続が再確立されしだいユーザーストアと同期されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定をここまたはINIファイルで構成しない場合、オフラインプロファイルサポート機能は無効になります。

## ユーザーストアへのパス

この設定項目では、ユーザーのレジストリなどの設定や同期対象のファイルを格納するユーザーストアのパスを指定します。

デフォルトでは、ホームドライブ上のWindowsフォルダーが使用されます。

この設定項目が無効な場合、ユーザー設定はホームフォルダーのWindowsサブフォルダーに格納されます。

以下のパスを設定できます。

- **相対パス**：ホームディレクトリ（通常Active Directoryユーザーの#homeDirectory#属性で構成される）からの相対パスです。
- **絶対UNCパス**：通常、サーバー共有またはDFS名前空間です。
- **無効化または未構成**：この場合、#homeDirectory#\Windowsの値が使用されます。

この設定項目を構成する場合、以下の変数を使用します。

- パーセントで囲まれたシステム環境変数（%ProfVer%など）。ただし、システム環境変数が正しくセットアップされている必要があります。
- ハッシュで囲まれたActive Directoryユーザーオブジェクトの属性（#sAMAccountName#など）。
- Profile managementの変数。詳しくは、Profile managementのドキュメントを参照してください。

%username%や%userdomain%などのユーザー環境変数を使用したり、locationやusersなどの組織の変数を完全に定義するカスタム属性を作成したりすることもできます。属性では大文字と小文字が区別されます。

例：

- 「\\server\share\#sAMAccountName#」と指定した場合、UNCパス\\server\share\JohnSmithにユーザー設定が格納されます（現在のユーザーの#sAMAccountName#属性がJohnSmithである場合）。
- 「\\server\profiles\$\%USERNAME%.%USERDOMAIN%\!CTX\_PROFILEVER!!CTX\_OSBITNESS!」と指定した場合、\\server\profiles\$\JohnSmith.DOMAINCONTROLLER1\v2x64などに格納されます。

**重要**：属性や変数を使用する場合は、NTUSER.DATがあるフォルダーの1つ上のフォルダーを指定していることを確認してください。たとえば、NTUSER.DATが\\server\profiles\$\JohnSmith.Finance\v2x64\UPM\_Profileにある場合は、ユーザーストアのパスとして「\\server\profiles\$\JohnSmith.Finance\v2x64」を指定します。UPM\_Profileサブフォルダーを含める必要はありません。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、ホームドライブ上のWindowsフォルダーが使用されます。

## ローカル管理者のログオン処理

この設定項目では、BUILTIN\Administratorsグループのメンバーのログオンを処理するかどうかを指定します。これにより、ローカルの管理者権限があるドメインユーザー（通常、仮想デスクトップが割り当てられているユーザー）がログオン時のプロファイル処理をバイパスして、Profile managementで問題が生じているデスクトップのトラブルシューティングを行えるようになります。

この設定項目が無効または未構成の場合、サーバーオペレーティングシステムではドメインユーザーのログオンは処理されませんが、ローカル管理者のログオンは処理されません。デスクトップオペレーティングシステムでは、ローカル管理者のログオンも処理されます。

この設定項目はデフォルトで無効になっており、ローカル管理者のログオンは処理されません。



この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、ローカル管理者のログオンは処理されません。

### 処理済みグループ

この設定項目では、Profile managementでのプロファイル処理の対象となるコンピューターのローカルグループおよびドメイングループ（ローカル、グローバル、およびユニバーサル）を指定します。

この設定項目を有効にすると、指定したユーザーグループのプロファイルのみがProfile managementで処理されるようになります。

この設定項目はデフォルトで無効になっており、すべてのユーザーグループのプロファイルが処理されます。

ドメイングループは、「<DOMAIN NAME>\<GROUP NAME>」形式で指定します。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、すべてのユーザーグループのプロファイルが処理されます。

# クロスプラットフォームのポリシー設定

Jan 04, 2017

[クロスプラットフォーム] カテゴリには、Profile managementのクロスプラットフォーム機能を構成するための設定項目が含まれています。

## クロスプラットフォーム設定ユーザーグループ

この設定項目では、クロスプラットフォーム設定機能が有効な場合にプロファイル処理するWindowsユーザーグループを指定します。

この設定項目はデフォルトで無効になっており、[処理済みグループ] 設定で指定されているすべてのユーザーグループのプロファイルが処理されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、すべてのユーザーグループのプロファイルが処理されます。

## クロスプラットフォーム設定の有効化

この設定項目では、複数のオペレーティングシステム上で同じアプリケーションを実行するユーザーのプロファイルの移行および移動を可能にするクロスプラットフォーム機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、クロスプラットフォーム設定は適用されません。

## クロスプラットフォーム定義へのパス

この設定項目では、ダウンロードパッケージからコピーされた定義ファイルのネットワーク上の場所をUNCパスとして指定します。

注：このパスには、ユーザーの読み取りアクセスおよび管理者の書き込みアクセスが設定されており、SMB (Server Message Block) またはCIFS (Common Internet File System) ファイル共有である必要があります。

デフォルトでは、パスは指定されていません。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、クロスプラットフォーム設定は適用されません。

## クロスプラットフォーム設定ストアへのパス

この設定項目では、クロスプラットフォーム設定ストアへのパスを指定します。クロスプラットフォーム設定ストアとは、ユーザーのクロスプラットフォーム設定を格納するフォルダーを指します。パスは、UNCパスまたはホームディレクトリからの相対パスで指定します。

注：クロスプラットフォーム設定ストアには、ユーザーの書き込みアクセスが設定されている必要があります。

この設定項目はデフォルトで無効になっており、Windows\PM\_CPが使用されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、デフォルト値が使用されます。

#### クロスプラットフォーム設定を作成するためのソース

この設定項目では、プラットフォーム組織単位 (OU)のプラットフォームを「基本プラットフォーム」として指定します。基本プラットフォームのプロファイルのデータは、クロスプラットフォーム設定ストアに移行されます。

各プラットフォームのプロファイルのセットは、個別のOUに格納されます。このため、管理者はどのプラットフォームのプロファイルデータをクロスプラットフォーム設定ストアに格納するかを決定する必要があります。このプラットフォームを「基本プラットフォーム」と呼びます。

この設定項目を有効にすると、クロスプラットフォーム設定ストアの定義ファイルにデータがない場合、またはキャッシュされた単一プラットフォームプロファイルのデータがストア内の定義データよりも新しい場合に、Profile managementが単一プラットフォームプロファイルのデータをストアに移行します。

**重要：**この設定項目を複数のOUやユーザー/マシンオブジェクトで有効にすると、最初のユーザーがログオンしたプラットフォームが基本プラットフォームになります。

この設定項目はデフォルトで無効になっており、単一プラットフォームプロファイルのデータはストアに移行されません。

# ファイルシステムのポリシー設定

Jan 04, 2017

[ファイルシステム] カテゴリには、プロファイルがインストールされているシステムとユーザーストア間で同期する、ユーザープロファイル内のファイルやフォルダーの構成に関する設定項目が含まれています。

# 除外のポリシー設定

Jan 04, 2017

[除外] カテゴリには、ユーザープロファイル内のファイルやディレクトリを同期処理から除外するための設定項目が含まれています。

## 除外の一覧 - ディレクトリ

この設定項目では、同期処理から除外する、ユーザープロファイル内のフォルダーを指定します。

フォルダー名は、ユーザープロファイル (%USERPROFILE%) からの相対パスで指定します。

この設定項目はデフォルトで無効になっており、ユーザープロファイル内のすべてのフォルダーが同期されます。

例：「Desktop」と指定した場合、ユーザープロファイルのデスクトップフォルダーは同期されません。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、ユーザープロファイル内のすべてのフォルダーが同期されます。

## 除外の一覧 - ファイル

この設定項目では、同期処理から除外する、ユーザープロファイル内のファイルを指定します。

この設定項目はデフォルトで無効になっており、ユーザープロファイル内のすべてのファイルが同期されます。

ファイル名は、ユーザープロファイル (%USERPROFILE%) からの相対パスで指定します。ワイルドカード文字を使用することもできます（再帰的に適用されます）。

例：「Desktop\Desktop.ini」と指定した場合、デスクトップフォルダーのDesktop.iniファイルは同期されません。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、ユーザープロファイル内のすべてのファイルが同期されます。

# 同期のポリシー設定

Jan 04, 2017

[同期] カテゴリには、プロファイルがインストールされているシステムとユーザーストア間で同期する、ユーザープロファイル内のファイルやフォルダーの指定に関する設定項目が含まれています。

## 同期するディレクトリ

この設定項目では、Profile managementで同期する、除外フォルダー内のファイルを指定します。デフォルトでは、ユーザープロファイル内のすべてのファイルがProfile managementにより同期されます。ユーザープロファイルのサブフォルダーは、この一覧に含めなくても同期されます。詳しくは、「[項目の包含および除外](#)」を参照してください。

この一覧にパスを追加するときは、ユーザープロファイルからの相対パスを入力します。

例：「Desktop\exclude\include」と指定した場合、Desktop\excludeフォルダーを同期対象から除外しても、includeフォルダーは同期されます。

この設定はデフォルトで無効になっており、フォルダーは指定されていません。

この設定をここで構成しない場合、INIファイルの値が使用されます。

この設定をここまたはINIファイルで構成しない場合、ユーザープロファイル内の非除外フォルダーのみが同期されます。

## 同期するファイル

この設定項目では、Profile managementで同期する、除外フォルダー内のファイルを指定します。デフォルトでは、ユーザープロファイル内のすべてのファイルがProfile managementにより同期されます。ユーザープロファイル内のファイルは、この一覧に含めなくても同期されます。詳しくは、「[項目の包含および除外](#)」を参照してください。

この一覧にパスを追加するときは、ユーザープロファイルからの相対パスを入力します。相対パスは、ユーザープロファイルの場所から相対的に解釈されます。ワイルドカード文字は、ファイル名に対してのみ使用できます。ワイルドカードは入れ子にできず、再帰的に適用されます。

例：

- 「AppData\Local\Microsoft\Office\Access.qat」と指定した場合、デフォルト構成で除外されるフォルダー内のファイルAccess.qatは同期されます。
- 「AppData\Local\MyApp\\*.cfg」と指定した場合、プロファイルフォルダーAppData\Local\MyAppとそのサブフォルダー内の.cfg拡張子を持つすべてのファイルが同期されます。

この設定はデフォルトで無効になっており、ファイルは指定されていません。

この設定をここで構成しない場合、INIファイルの値が使用されます。

この設定をここまたはINIファイルで構成しない場合、ユーザープロファイル内の非除外ファイルのみが同期されます。

## ミラーリングするフォルダー

この設定項目では、ミラーリングするフォルダーを、ユーザープロファイルのルートフォルダーからの相対パスで指定します。この設定項目を構成することで、相互に依存するファイルが格納されたトランザクションフォルダー（参照フォルダー）に関する問題を解決できます。

フォルダーのミラーリングにより、Profile managementがトランザクションフォルダーおよびその内容を単一エンティティ

として処理するため、プロファイルの膨張を防ぐことができます。ただし、ミラーリングされたフォルダー内のファイルが複数のセッションで変更された場合、最後の更新によりそのファイルが上書きされ、プロファイルの変更が失われることがあります。点に注意してください。

たとえば、Internet ExplorerのCookieフォルダーをミラーリングして、Index.datが対象のCookieと同期されるように設定できます。

ユーザーが異なるサーバー上の2つのInternet Explorerセッションを実行して、各セッションで異なるWebサイトにアクセスする場合、それらのWebサイトからのCookieがそれぞれのサーバーに追加されます。ユーザーが1つ目のセッションからログオフするときに（アクティブライトバック機能が有効な場合はセッションの途中で）、2つ目のセッションからのCookieにより最初のセッションのCookieが置き換えられなければなりません。ところが、これらのCookieはマージされてしまい、Index.datのCookieへの参照は最新ではなくなります。新しいセッションでの以降のWebサイト閲覧ではマージが繰り返され、Cookieフォルダーのサイズが膨張します。

Cookieフォルダーをミラーリングすると、ユーザーがログオフするたびにCookieが最新セッションのもので上書きされ、Index.datが最新の状態で維持されます。

この設定項目はデフォルトで無効になっており、フォルダーはミラーリングされません。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、フォルダーはミラーリングされません。

# フォルダーリダイレクトのポリシー設定

Jan 04, 2017

[フォルダーリダイレクト] カテゴリには、プロファイル内の一般的なフォルダーをネットワーク上の共有フォルダーにリダイレクトするための設定項目が含まれています。

## 管理者アクセスを許可

この設定項目では、リダイレクトされたユーザーのフォルダーに管理者がアクセスすることを有効または無効にします。

この設定項目はデフォルトで無効になっており、リダイレクトされたフォルダーの内容に対してユーザーの排他アクセスが付与されています。

## ドメイン名を包含

この設定項目では、リダイレクトされるフォルダーのUNCパスに環境変数%userdomain%を含めることを有効または無効にします。

この設定項目はデフォルトで無効になっており、リダイレクトされるフォルダーのUNCパスに環境変数%userdomain%は含まれません。



# AppData (Roaming) のポリシー設定

Jan 04, 2017

[AppData (Roaming) ] カテゴリには、ユーザープロファイルのAppData (Roaming) フォルダをネットワーク上の共有フォルダにリダイレクトするための設定項目が含まれています。

## AppData (Roaming) パス

この設定項目では、AppData (Roaming) フォルダのリダイレクト先のネットワークフォルダを指定します。

この設定項目はデフォルトで無効になっており、リダイレクト先は指定されていません。

この設定項目が未構成の場合、このフォルダはProfile managementによりリダイレクトされません。

## AppData (Roaming) のリダイレクト設定

この設定項目では、AppData (Roaming) フォルダの内容をどのようにリダイレクトするかを指定します。

デフォルトでは、UNCパスにリダイレクトされます。

この設定項目が未構成の場合、このフォルダはProfile managementによりリダイレクトされません。

# アドレス帳のポリシー設定

Jan 04, 2017

[アドレス帳] カテゴリには、ユーザープロファイルのアドレス帳フォルダーをネットワーク上の共有フォルダーにリダイレクトするための設定項目が含まれています。

## アドレス帳パス

この設定項目では、アドレス帳フォルダーのリダイレクト先のネットワークフォルダーを指定します。

この設定項目はデフォルトで無効になっており、リダイレクト先は指定されていません。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

## アドレス帳のリダイレクト設定

この設定項目では、アドレス帳フォルダーの内容をどのようにリダイレクトするかを指定します。

デフォルトでは、UNCパスにリダイレクトされます。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

# デスクトップのポリシー設定

Jan 04, 2017

[デスクトップ] カテゴリには、ユーザープロファイルのデスクトップフォルダーをネットワーク上の共有フォルダーにリダイレクトするための設定項目が含まれています。

## デスクトップパス

この設定項目では、デスクトップフォルダーのリダイレクト先のネットワークフォルダーを指定します。

この設定項目はデフォルトで無効になっており、リダイレクト先は指定されていません。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

## デスクトップのリダイレクト設定

この設定項目では、デスクトップフォルダーの内容をどのようにリダイレクトするかを指定します。

デフォルトでは、UNCパスにリダイレクトされます。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

# ドキュメントのポリシー設定

Jan 04, 2017

[ドキュメント] カテゴリには、ユーザープロファイルのドキュメントフォルダーをネットワーク上の共有フォルダーにリダイレクトするための設定項目が含まれています。

## ドキュメントパス

この設定項目では、ドキュメントフォルダーのリダイレクト先のネットワークフォルダーを指定します。

この設定項目はデフォルトで無効になっており、リダイレクト先は指定されていません。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

ファイルをドキュメントフォルダーにリダイレクトするだけでなく、ミュージック、ピクチャ、およびビデオフォルダーにもリダイレクトするため、[ドキュメントパス] 設定を有効にする必要があります。

## ドキュメントのリダイレクト設定

この設定項目では、ドキュメントフォルダーの内容をどのようにリダイレクトするかを指定します。

デフォルトでは、UNCパスにリダイレクトされます。

ドキュメントフォルダーのリダイレクト方法として、以下のいずれかのオプションを選択します。

- 次のUNCパスにリダイレクト: [ドキュメントパス] 設定で指定されたUNCパスにリダイレクトします。
- ユーザーのホームディレクトリにリダイレクト: ユーザーのホームディレクトリ (通常Active Directoryでユーザーの #homeDirectory#属性として構成される) にリダイレクトします。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

# ダウンロードのポリシー設定

Jan 04, 2017

[ダウンロード] カテゴリには、ユーザープロファイルのダウンロードフォルダーをネットワーク上の共有フォルダーにリダイレクトするための設定項目が含まれています。

## ダウンロードパス

この設定項目では、ダウンロードフォルダーのリダイレクト先のネットワークフォルダーを指定します。

この設定項目はデフォルトで無効になっており、リダイレクト先は指定されていません。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

## ダウンロードのリダイレクト設定

この設定項目では、ダウンロードフォルダーの内容をどのようにリダイレクトするかを指定します。

デフォルトでは、UNCパスにリダイレクトされます。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

# お気に入りのポリシー設定

Jan 04, 2017

【お気に入り】カテゴリには、ユーザープロファイルのお気に入りフォルダーをネットワーク上の共有フォルダーにリダイレクトするための設定項目が含まれています。

## お気に入りパス

この設定項目では、お気に入りフォルダーのリダイレクト先のネットワークフォルダーを指定します。

この設定項目はデフォルトで無効になっており、リダイレクト先は指定されていません。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

## お気に入りのリダイレクト設定

この設定項目では、お気に入りフォルダーの内容をどのようにリダイレクトするかを指定します。

デフォルトでは、UNCパスにリダイレクトされます。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

# リンクのポリシー設定

Jan 04, 2017

[リンク] カテゴリには、ユーザープロファイルのリンクフォルダーをネットワーク上の共有フォルダーにリダイレクトするための設定項目が含まれています。

## リンクパス

この設定項目では、リンクフォルダーのリダイレクト先のネットワークフォルダーを指定します。

この設定項目はデフォルトで無効になっており、リダイレクト先は指定されていません。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

## リンクのリダイレクト設定

この設定項目では、リンクフォルダーの内容をどのようにリダイレクトするかを指定します。

デフォルトでは、UNCパスにリダイレクトされます。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

# ミュージックのポリシー設定

Jan 04, 2017

[ミュージック] カテゴリには、ユーザープロファイルのミュージックフォルダーをネットワーク上の共有フォルダーにリダイレクトするための設定項目が含まれています。

## ミュージックパス

この設定項目では、ミュージックフォルダーのリダイレクト先のネットワークフォルダーを指定します。

この設定項目はデフォルトで無効になっており、リダイレクト先は指定されていません。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

## ミュージックのリダイレクト設定

この設定項目では、ミュージックフォルダーの内容をどのようにリダイレクトするかを指定します。

デフォルトでは、UNCパスにリダイレクトされます。

ミュージックフォルダーのリダイレクト方法として、以下のいずれかのオプションを選択します。

- 次のUNCパスにリダイレクト: [ミュージックパス] 設定で指定されたUNCパスにリダイレクトします。
- ドキュメントフォルダーに相対的リダイレクト: ドキュメントフォルダーのリダイレクト先と相対的に同じ場所にあるフォルダーにリダイレクトします。

コンテンツをドキュメントフォルダーに相対するフォルダーに梨諾冷雨とするには、[ドキュメントパス設定を有効にする必要があります]。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。



# ピクチャのポリシー設定

Jan 04, 2017

[ピクチャ] カテゴリには、ユーザープロファイルのピクチャフォルダーをネットワーク上の共有フォルダーにリダイレクトするための設定項目が含まれています。

## ピクチャパス

この設定項目では、ピクチャフォルダーのリダイレクト先のネットワークフォルダーを指定します。

この設定項目はデフォルトで無効になっており、リダイレクト先は指定されていません。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

## ピクチャのリダイレクト設定

この設定項目では、ピクチャフォルダーの内容をどのようにリダイレクトするかを指定します。

デフォルトでは、UNCパスにリダイレクトされます。

ピクチャフォルダーのリダイレクト方法として、以下のいずれかのオプションを選択します。

- 次のUNCパスにリダイレクト: [ピクチャパス] 設定で指定されたUNCパスにリダイレクトします。
- ドキュメントフォルダーに相対的リダイレクト: ドキュメントフォルダーのリダイレクト先と相対的に同じ場所にあるフォルダーにリダイレクトします。

コンテンツをドキュメントフォルダーに相対するフォルダーに梨諾冷雨とするには、[ドキュメントパス設定を有効にする] 必要があります。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

# 保存したゲームのポリシー設定

Jan 04, 2017

〔保存したゲーム〕カテゴリには、ユーザープロファイルにある保存したゲームフォルダーをネットワーク上の共有フォルダーにリダイレクトするための設定項目が含まれています。

## 保存したゲームのリダイレクト設定

この設定項目では、保存したゲームフォルダーの内容をどのようにリダイレクトするかを指定します。

デフォルトでは、UNCパスにリダイレクトされます。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

## 保存したゲームパス

この設定項目では、保存したゲームフォルダーのリダイレクト先のネットワークフォルダーを指定します。

この設定項目はデフォルトで無効になっており、リダイレクト先は指定されていません。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

# 検索のポリシー設定

Jan 04, 2017

[検索] カテゴリには、ユーザープロファイルの検索フォルダーをネットワーク上の共有フォルダーにリダイレクトするための設定項目が含まれています。

## 検索のリダイレクト設定

この設定項目では、検索フォルダーの内容をどのようにリダイレクトするかを指定します。

デフォルトでは、UNCパスにリダイレクトされます。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

## 検索パス

この設定項目では、検索フォルダーのリダイレクト先のネットワークフォルダーを指定します。

この設定項目はデフォルトで無効になっており、リダイレクト先は指定されていません。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

# スタートメニューのポリシー設定

Jan 04, 2017

[スタートメニュー] カテゴリには、ユーザープロファイルのスタートメニューフォルダーをネットワーク上の共有フォルダーにリダイレクトするための設定項目が含まれています。

## スタートメニューのリダイレクト設定

この設定項目では、スタートメニューフォルダーの内容をどのようにリダイレクトするかを指定します。

デフォルトでは、UNCパスにリダイレクトされます。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

## スタートメニューパス

この設定項目では、スタートメニューフォルダーのリダイレクト先のネットワークフォルダーを指定します。

この設定項目はデフォルトで無効になっており、リダイレクト先は指定されていません。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

# ビデオのポリシー設定

Jan 04, 2017

[ビデオ] カテゴリには、ユーザープロファイルのビデオフォルダーをネットワーク上の共有フォルダーにリダイレクトするための設定項目が含まれています。

## ビデオのリダイレクト設定

この設定項目では、ビデオフォルダーの内容をどのようにリダイレクトするかを指定します。

デフォルトでは、UNCパスにリダイレクトされます。

ビデオフォルダーのリダイレクト方法として、以下のいずれかのオプションを選択します。

- 次のUNCパスにリダイレクト: [ビデオパス] 設定で指定されたUNCパスにリダイレクトします。
- ドキュメントフォルダーに相対的リダイレクト: ドキュメントフォルダーのリダイレクト先と相対的に同じ場所にあるフォルダーにリダイレクトします。

コンテンツをドキュメントフォルダーに相対するフォルダーに梨諾冷雨とするには、[ドキュメントパス設定を有効にする必要があります]。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

## ビデオパス

この設定項目では、ビデオフォルダーのリダイレクト先のネットワークフォルダーを指定します。

この設定項目はデフォルトで無効になっており、リダイレクト先は指定されていません。

この設定項目が未構成の場合、このフォルダーはProfile managementによりリダイレクトされません。

# ログのポリシー設定

Jan 04, 2017

[ログ] カテゴリには、Profile managementのログ機能の構成に関する設定項目が含まれています。

## Active Directory操作

この設定項目では、Active Directoryで実行された操作についての詳細なログ機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目を有効にするときは、[ログの有効化] 設定で [有効] が選択されていることを確認してください。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、エラーおよび一般情報のみがログに記録されます。

## 一般的な情報

この設定項目では、一般的な情報についての詳細なログ機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目を有効にするときは、[ログの有効化] 設定で [有効] が選択されていることを確認してください。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、エラーおよび一般情報のみがログに記録されます。

## 一般的な警告

この設定項目では、一般的な警告についての詳細なログ機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目を有効にするときは、[ログの有効化] 設定で [有効] が選択されていることを確認してください。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、エラーおよび一般情報のみがログに記録されます。

## ログの有効化

この設定項目では、Profile managementのデバッグモード（詳細ログモード）のログ機能を有効または無効にします。デバッグモードでは、詳細な状態情報が%SystemRoot%\System32\Logfiles\UserProfileManagerフォルダーのログファイルに記録されます。

この設定項目はデフォルトで無効になっており、エラーのみがログに記録されます。

この設定項目は、Profile managementのトラブルシューティング時にのみ有効にすることをお勧めします。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、エラーのみがログに記録されます。

## ファイルシステム操作

この設定項目では、ファイルシステムで実行された操作についての詳細なログ機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目を有効にするときは、[ログの有効化] 設定で [有効] が選択されていることを確認してください。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、エラーおよび一般情報のみがログに記録されます。

## ファイルシステム通知

この設定項目では、ファイルシステムで発生した通知についての詳細なログ機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目を有効にするときは、[ログの有効化] 設定で [有効] が選択されていることを確認してください。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、エラーおよび一般情報のみがログに記録されます。

## ログオフ

この設定項目では、ユーザーのログオフについての詳細なログ機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目を有効にするときは、[ログの有効化] 設定で [有効] が選択されていることを確認してください。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、エラーおよび一般情報のみがログに記録されます。

## ログオン

この設定項目では、ユーザーのログオンについての詳細なログ機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目を有効にするときは、[ログの有効化] 設定で [有効] が選択されていることを確認してください。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、エラーおよび一般情報のみがログに記録されます。

## ログファイルの最大サイズ

この設定項目では、Profile managementで生成されるログファイルの最大サイズをバイト単位で指定します。

デフォルトでは、1048576バイト (1MB) に設定されています。

ディスクに十分な空き領域がある場合は、5MB以上を指定することをお勧めします。ログファイルのサイズがここで指定した値を超えると、既存のバックアップファイル (.bak) が削除され、そのログファイルがバックアップファイルとして保存されて新しいログファイルが作成されます。

ログファイルは、%SystemRoot%\System32\Logfiles\UserProfileManagerフォルダーに生成されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、デフォルト値が使用されます。

#### ログファイルへのパス

この設定項目では、Profile managementのログファイルの保存フォルダーを指定します。

この設定項目はデフォルトで無効になっており、デフォルトのフォルダー (%SystemRoot%\System32\Logfiles\UserProfileManager) にログファイルが生成されます。

保存フォルダーのパスとして、ローカルドライブ、リモートドライブ、またはネットワークドライブ (UNCパス) を指定できます。リモートドライブは大規模な分散環境では便利ですが、大量のネットワークトラフィックが発生するためログファイルには不適切である場合があります。プロビジョニングした仮想マシンに永続的なハードドライブがある場合は、そのドライブ上のローカルパスを指定します。これにより、仮想マシンを再起動してもログファイルが保持されます。永続的なハードドライブがない仮想マシンの場合、UNCパスを指定するとログファイルを保持できますが、この仮想マシンのシステムアカウントにはそのUNCパスに対する書き込みアクセス権が必要です。オフラインプロファイル機能で管理するラップトップコンピューターの場合は、ローカルパスを使用します。

ログファイルをUNCパス上のフォルダーに保存する場合は、そのフォルダーに適切なアクセス制御リストを適用して、認証されたユーザーやコンピューターのみがログファイルにアクセスできるようにすることをお勧めします。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、デフォルト値である%SystemRoot%\System32\Logfiles\UserProfileManagerフォルダーが使用されます。

#### 個人用ユーザー情報

この設定項目では、個人用ユーザー情報についての詳細なログ機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目を有効にするときは、[ログの有効化] 設定で [有効] が選択されていることを確認してください。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、エラーおよび一般情報のみがログに記録されます。

#### ログオンおよびログオフ時のポリシー値

この設定項目では、ユーザーのログオン時およびログオフ時のポリシー設定値についての詳細なログ機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目を有効にするときは、[ログの有効化] 設定で [有効] が選択されていることを確認してください。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、エラーおよび一般情報のみがログに記録されます。

#### レジストリ操作



この設定項目では、レジストリで実行された操作についての詳細なログ機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目を有効にするときは、[ログの有効化] 設定で [有効] が選択されていることを確認してください。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、エラーおよび一般情報のみがログに記録されます。

#### ログオフ時のレジストリ差分

この設定項目では、ユーザーのログオフ時のレジストリ設定の相違についての詳細なログ機能を有効または無効にします。

デフォルトでは、無効になっています。

この設定項目を有効にするときは、[ログの有効化] 設定で [有効] が選択されていることを確認してください。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、エラーおよび一般情報のみがログに記録されます。

# プロファイル制御のポリシー設定

Jan 04, 2017

[プロファイル制御] カテゴリには、Profile managementでのユーザープロファイル管理機能を制御するための設定項目が含まれています。

## キャッシュしたプロファイルを削除する前の待ち時間

この設定項目では、ローカルにキャッシュされたプロファイルをそのユーザーのログオフ後にProfile managementが削除するまでの待機時間を指定します。

0を指定すると、ログオフ処理が完了した後でプロファイルが直ちに削除されます。Profile managementでは、1分ごとにログオフの状態がチェックされます。このため、この設定項目で60を指定すると、ユーザーのログオフ後1~2分後にプロファイルが削除されます。ログオフ時にファイルやレジストリハブにアクセスするプロセスがある場合は、ここで待機時間を延長できます。また、プロファイルのサイズが大きい場合、待機時間を延長することでログオフ時間が短縮されることがあります。

デフォルトでは0が指定されており、ローカルにキャッシュされたプロファイルがログオフ後に直ちに削除されます。

この設定項目を有効にするときは、[ログオフ時にローカルでキャッシュしたプロファイルの削除]設定で[有効]が選択されていることを確認してください。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、プロファイルは直ちに削除されます。

## ログオフ時にローカルでキャッシュしたプロファイルの削除

この設定項目では、ユーザーのログオフ後にローカルにキャッシュされたプロファイルを削除するかどうかを指定します。

この設定項目を有効にすると、ユーザーのローカルプロファイルキャッシュがログオフ後に削除されます。ターミナルサーバーではこの設定項目を有効にすることをお勧めします。

この設定項目はデフォルトで無効になっており、ローカルプロファイルはユーザーのログオフ後も保持されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、キャッシュされたプロファイルは削除されません。

## ローカルプロファイル競合の制御

この設定項目では、ユーザーストアのプロファイルとローカルのWindowsユーザープロファイルの両方が存在する場合に、Profile managementがどのように動作するかを指定します。

デフォルトでは、Profile managementはローカルのWindowsプロファイルを使用しますが、そのプロファイルを変更することはありません。

Profile managementの動作を制御するには、次のいずれかのオプションを選択します。

- ローカルプロファイルを使用: Profile managementはローカルのプロファイルを使用し、そのプロファイルを変更することはありません。
- ローカルプロファイルを削除: Profile managementは、ローカルのWindowsユーザープロファイルを削除して、ユーザーストアからCitrixユーザープロファイルをインポートします。

- ローカルプロファイル名を変更: Profile managementは、ローカルのWindowsユーザープロファイルの名前を変更してバックアップとして保持し、ユーザーストアからCitrixユーザープロファイルをインポートします。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、既存のローカルプロファイルを使用します。

### 既存のプロファイルの移行

この設定項目では、ログオンしたユーザーのプロファイルがユーザーストアに存在しない場合に、どのプロファイルをユーザーストアに移行するかを指定します。

Profile managementでは、ユーザーストアにプロファイルが存在しないユーザーがログオンしたときに、既存のプロファイルが自動的にユーザーストアに移行されます。移行が完了すると、現在のセッション、および同じユーザーストアのパスが作成されたすべてのセッションで、ユーザーストアのプロファイルがProfile managementで使用されます。

デフォルトでは、ローカルプロファイルおよび移動プロファイルがログオン時にユーザーストアに移行されます。

移行されるプロファイルを指定するには、以下のいずれかのオプションを選択します。

- ローカルおよび移動
- ローカル
- 移動
- なし（無効）

[なし]を選択すると、通常のWindowsの動作（つまりProfile managementがインストールされていない場合の動作）に基づいて新しいプロファイルが作成されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、既存のローカルプロファイルおよび移動プロファイルが移行されます。

### テンプレートプロファイルへのパス

この設定項目では、Profile managementで新しいユーザープロファイルを作成するときにテンプレートとして使用するプロファイルのパスを指定します。

このパスは、NTUSER.DATレジストリファイルや、テンプレートプロファイルに必要なそのほかのフォルダーやファイルを格納しているフォルダーのものである必要があります。

注：パスに「NTUSER.DAT」を含めないでください。たとえば、「\\myservername\myprofiles\template\ntuser.dat」ではなく「\\myservername\myprofiles\template」を指定します。

UNCパスやローカルマシン上のパスなどの絶対パスを使用します。たとえば、Citrix Provisioning Servicesイメージ上のテンプレートプロファイルを永続的に指定するにはローカルマシン上のパスを指定します。相対パスは使用できません。

注：Active Directory属性の拡張、システム環境変数、および%USERNAME%や%USERDOMAIN%変数を使用することはできません。

この設定項目はデフォルトで無効になっており、最初にログオンしたデバイス上のデフォルトのユーザープロファイルを基にそのユーザーのプロファイルが作成されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、テンプレートは使用されません。

### テンプレートプロファイルがローカルプロファイルを上書きする

この設定項目では、新しいユーザープロファイルの作成時にローカルプロファイルよりもテンプレートプロファイルを優先する機能を有効または無効にします。

デフォルトでは、ユーザーにCitrixユーザープロファイルがなく、ローカルのWindowsユーザープロファイルが存在する場合、デフォルトでローカルのプロファイルが使用され、ユーザーストアに移行されます。この設定項目を有効にすると、新しいユーザープロファイルの作成時にローカルプロファイルではなくテンプレートプロファイルが使用されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、テンプレートは使用されません。

#### テンプレートプロファイルが移動プロファイルを上書きする

この設定項目では、新しいユーザープロファイルの作成時に移動プロファイルよりもテンプレートプロファイルを優先する機能を有効または無効にします。

デフォルトでは、ユーザーにCitrixユーザープロファイルがなく、Windowsの移動ユーザープロファイルが存在する場合、デフォルトで移動プロファイルが使用され、ユーザーストアに移行されます。この設定項目を有効にすると、新しいユーザープロファイルの作成時に移動プロファイルではなくテンプレートプロファイルが使用されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、テンプレートは使用されません。

#### すべてのログオンでCitrix固定プロファイルとして使用されるテンプレートプロファイル

この設定項目では、Profile managementで新しいユーザープロファイルを作成するときに、テンプレートプロファイルをデフォルトのプロファイルとして使用するかどうかを指定します。

この設定項目はデフォルトで無効になっており、最初にログオンしたデバイス上のデフォルトのユーザープロファイルを基にそのユーザーのプロファイルが作成されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、テンプレートは使用されません。

# レジストリのポリシー設定

Jan 04, 2017

[レジストリ] カテゴリには、特定のレジストリキーをProfile managementの処理対象として指定したり除外したりするための設定項目が含まれています。

## 除外の一覧

この設定項目では、ユーザーのログオフ時にProfile managementの処理対象から除外するHKEY\_CURRENT\_USERハイブのレジストリキーを指定します。

この設定項目を有効にすると、一覧に追加したレジストリキーがユーザーのログオフ時にProfile managementで処理されなくなります。

この設定項目はデフォルトで無効になっており、HKEY\_CURRENT\_USERハイブのすべてのレジストリキーがユーザーのログオフ時に処理されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、HKEY\_CURRENT\_USERハイブのすべてのレジストリキーがユーザーのログオフ時に処理されます。

## 包含の一覧

この設定項目では、ユーザーのログオフ時にProfile managementの処理対象にするHKEY\_CURRENT\_USERハイブのレジストリキーを指定します。

この設定項目を有効にすると、一覧に追加したレジストリキーのみがユーザーのログオフ時にProfile managementで処理されます。

この設定項目はデフォルトで無効になっており、HKEY\_CURRENT\_USERハイブのすべてのレジストリキーがユーザーのログオフ時に処理されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、HKEY\_CURRENT\_USERハイブのすべてのレジストリキーがユーザーのログオフ時に処理されます。

# ストリーム配信ユーザープロファイルのポリシー設定

Jan 04, 2017

[ストリーム配信ユーザープロファイル] カテゴリには、Profile managementでのストリーム配信ユーザープロファイル管理機能を制御するための設定項目が含まれています。

## 常時キャッシュ

この設定項目では、ユーザーのログオン後にストリーム配信されたファイルをキャッシュするかどうかを指定します。ファイルをキャッシュするとネットワークの帯域幅消費が減少し、ユーザーエクスペリエンスが向上します。

この設定項目は、[プロファイルストリーム配信] 設定と一緒に使用します。

この設定項目はデフォルトで無効になっており、ユーザーのログオン後にストリーム配信されたファイルはキャッシュされません。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、この機能が無効になります。

## 常時キャッシュサイズ

この設定項目では、ストリーム配信されるファイルの最小サイズをメガバイト (MB) 単位で指定します。Profile managementでは、ここで指定した値以上のサイズのファイルがユーザーのログオン後にキャッシュされます。

デフォルトでは0が指定されており、プロファイル全体がキャッシュされます。この場合、ユーザーのログオン後、バックグラウンドタスクとしてユーザーストアのプロファイルの内容すべてがProfile managementによりキャッシュされます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、この機能が無効になります。

## プロファイルストリーム配信

この設定項目では、Profile managementによるユーザープロファイルのストリーム配信機能を有効または無効にします。この設定項目を有効にすると、プロファイルに含まれるファイルやフォルダーが、ログオンしたユーザーがアクセスした時点でユーザーストアからローカルコンピューターに取得されます。待機領域内のレジストリエントリやファイルは、直ちに取得されます。

デフォルトでは、無効になっています。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、この機能が無効になります。

## ストリーム配信ユーザープロファイルグループ

この設定項目では、ストリーム配信する組織単位のユーザープロファイルをWindowsユーザーグループで指定します。

この設定項目を有効にすると、指定したユーザーグループのユーザープロファイルのみがストリーム配信されます。ほかのユーザープロファイルは、通常どおりに処理されます。

この設定項目はデフォルトで無効になっており、組織単位のすべてのユーザープロファイルが通常どおりに処理されます。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、すべてのユーザープロファイルが処理されます。

#### 待機領域のロックファイルのタイムアウト

この設定項目では、サーバーが応答不能になってユーザーストアのロックが解除されない場合に、待機領域のファイルをユーザーストアに同期するまでの日数を指定します。これにより、待機領域が膨張することを防いで、ユーザーストアに常に最新のファイルが同期されるようになります。

デフォルトでは、1日に設定されています。

この設定項目が未構成の場合、INIファイルの値が使用されます。

この設定項目およびINIファイルが未構成の場合、デフォルト値が使用されます。

# Receiverのポリシー設定

Jan 04, 2017

[Receiver] カテゴリには、仮想デスクトップ上のReceiver for Windowsで使用されるStoreFrontアドレスの一覧を構成するための設定項目が含まれています。

## StoreFrontアカウント一覧

この設定項目では、仮想デスクトップ上で実行されるReceiver for Windowsに設定するStoreFrontストアの一覧を指定します。デリバリーグループを作成するときに、管理者はこの一覧からストアを選択して、そのデリバリーグループの仮想デスクトップ上で動作するReceiver for Windowsに適用できます。

デフォルトでは、ストアは指定されていません。

一覧にストアを追加するときは、以下の情報をセミコロンで区切って入力します。

- ストア名：ユーザーに表示されるストア名です。
- ストアURL：ストアのURLです。
- ストアの有効/無効：そのストアをユーザーが使用できるかどうかを「On」または「Off」で指定します。
- ストアの説明：ユーザーに表示される説明です。

たとえば、「Sales Store;https://sales.mycompany.com/Citrix/Store/discovery;On;Store for Sales staff」のように入力します。



# Virtual Delivery Agentのポリシー設定

Jan 04, 2017

[Virtual Delivery Agent設定] カテゴリには、Virtual Delivery Agent (VDA) とController間の通信を制御するための設定項目が含まれています。

重要：VDAをDelivery Controllerに登録するときに、これらの設定項目で提供される情報が必要になります（自動更新機能を使用しない場合）。これらの情報は登録に必要であるため、グループポリシーエディターを使って以下の設定項目を構成する必要があります（VDAのインストール時にこれらの情報を指定する場合を除く）。

- コントローラー登録のIPv6ネットマスク
- コントローラー登録ポート
- コントローラーSID
- コントローラー
- IPv6コントローラー登録のみを使用する
- サイトGUID

## コントローラー登録のIPv6ネットマスク

この設定項目では、VDAで使用されるサブネットを指定できます。この場合、グローバルIPは使用されません。これにより、指定したIPv6アドレスおよびネットワークでのみVDAが登録されます。VDAは、指定されたネットマスクに最初にマッチしたアドレスでのみ登録されます。この設定項目を使用する場合は、[IPv6コントローラー登録のみを使用する] 設定を有効にする必要があります。

デフォルトでは、空白になっています。

## コントローラー登録ポート

この設定項目を使用する場合は、[コントローラーの自動更新を有効にする] 設定で [禁止] が選択されていることを確認してください。

この設定項目では、VDAのController登録をレジストリで行うときに使用されるTCP/IPポート番号を指定します。

デフォルトのポート番号は、80に設定されています。

## コントローラーSID

この設定項目を使用する場合は、[コントローラーの自動更新を有効にする] 設定で [禁止] が選択されていることを確認してください。

この設定項目では、VDAのController登録をレジストリで行うときに使用されるControllerのセキュリティ識別子 (SID) をスペース区切りの一覧で指定します。これはオプションの設定項目で、[Controller] 設定と一緒に使用して、登録に使用されるControllerの一覧を制限できます。

デフォルトでは、空白になっています。

## コントローラー

この設定項目を使用する場合は、[コントローラーの自動更新を有効にする] 設定で [禁止] が選択されていることを確認してください。

この設定項目では、VDAのController登録をレジストリで行うときに使用されるControllerの完全修飾ドメイン名 (FQDN) をスペース区切りの一覧で指定します。これはオプションの設定項目で、[コントローラーSID] 設定と一緒に使用することも

できます。

デフォルトでは、空白になっています。

### コントローラーの自動更新を有効にする

この設定項目では、インストール後のVDAをControllerに自動的に登録する機能を許可または禁止します。

VDAをControllerに登録すると、登録先のControllerにより環境内のControllerのFQDNおよびSIDの一覧がVDAに送信されます。この一覧の内容は、VDAにより永続的なストレージに書き込まれます。また、各Controllerは90分ごとにサイトのデータベースにアクセスして、Controllerの追加や削除、およびポリシーの変更内容について確認し、登録したVDAに更新情報を送信します。VDAは、受信した最新の一覧に基づいてすべてのControllerからの接続を受け入れます。

デフォルトでは、有効になっています。

### IPv6コントローラー登録のみを使用する

この設定項目では、Controllerへの登録時にVDAで使用されるアドレスの形式を指定します。

- この設定項目を有効にすると、そのマシンのIPv6アドレスを使用してVDAがControllerと登録および通信を行います。VDAがControllerと通信するときに、グローバルIPアドレス、ユニークローカルアドレス (ULA)、リンクローカルアドレス (ほかのIPv6アドレスを使用できない場合のみ) の順でIPv6アドレスが選択されます。
- この設定項目が無効な場合、そのマシンのIPv4アドレスを使用してVDAがControllerと登録および通信を行います。

デフォルトでは、無効になっています。

### サイトGUID

この設定項目を使用する場合は、[コントローラーの自動更新を有効にする]設定で[禁止]が選択されていることを確認してください。

この設定項目では、VDAのController登録をActive Directoryベースで行うときに使用される、サイトのグローバル一意識別子 (GUID) を指定します。

デフォルトでは、空白になっています。

# HDX 3D Proのポリシー設定

Jan 04, 2017

[HDX3DPro] カテゴリには、ユーザーの画質構成ツールを有効にして構成するための設定項目が含まれています。ユーザーがこのツールを使用すると、画質と応答性間のバランスをリアルタイムで調整して、帯域幅の使用を最適化できます。

## 無損失を有効にする

この設定項目では、ユーザーが画質構成ツールで無損失圧縮を有効にしたり無効にしたりすることを許可するかどうかを指定します。デフォルトでは、ユーザーは無損失圧縮を有効にできません。

ユーザーが無損失圧縮を有効にすると、自動的に画質構成ツールで設定可能な最高画質に設定されます。デフォルトでは、ユーザーデバイスとホストコンピューターの能力に応じて、GPUまたはCPUベースの圧縮が使用されます。

## HDX 3D Pro品質レベル

この設定項目では、ユーザーが画質構成ツールで設定できる画質調整範囲の最小値および最大値を指定します。

画質は0~100の値で指定します。最大値には、最小値を超える値を設定する必要があります。

# 仮想IPのポリシー設定

Jan 04, 2017

[仮想IP] カテゴリには、セッションの仮想ループバックアドレスの使用を制御するための設定項目が含まれています。

## 仮想IPループバックサポート

この設定項目では、各セッション固有の仮想ループバックアドレスの使用を有効にするかどうかを指定します。無効にすると、セッション固有の仮想ループバックアドレスは使用されません。

デフォルトでは、無効になっています。

## 仮想IPループバックプログラム一覧

この設定項目では、仮想ループバックアドレスを使用できるアプリケーション実行可能ファイルを指定します。一覧にプログラムを追加するときは、実行可能ファイルの名前のみを指定します。パス全体を入力する必要はありません。

デフォルトでは、実行可能ファイルは指定されていません。

# レジストリを使ったCOMポートおよびLPTポートリダイレクト設定の構成

Jan 04, 2017

COMポートおよびLPTポートのリダイレクト設定は、VDAイメージまたはマシンのレジストリ HKEY\_LOCAL\_MACHINE\Software\Citrix\GroupPolicy\Defaults\Deprecatedで構成します。

COMポートおよびLPTポートリダイレクトを有効にするには、以下のレジストリキーを追加してREG\_DWORD値を設定します。

注意：レジストリエディタの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディタの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディタは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

| レジストリキー                   | 説明                                        | 値                 |
|---------------------------|-------------------------------------------|-------------------|
| AllowComPortRedirection   | COMポートリダイレクトを許可または禁止する                    | 1 (許可) または 0 (禁止) |
| LimitComBw                | COMポートリダイレクトチャンネルの最大帯域幅                   | 数値                |
| LimitComBWPercent         | COMポートのリダイレクトチャンネルで使用可能な帯域幅のセッション全体に対する割合 | 0~100の数値          |
| AutoConnectClientComPorts | ユーザーデバイス側のCOMポートへの自動接続                    | 1 (許可) または 0 (禁止) |
| AllowLptPortRedirection   | LPTポートリダイレクトを許可または禁止する                    | 1 (許可) または 0 (禁止) |
| LimitLptBw                | LPTポートリダイレクトチャンネルの最大帯域幅                   | 数値                |
| LimitLptBwPercent         | LPTポートのリダイレクトチャンネルで使用可能な帯域幅のセッション全体に対する割合 | 0~100の数値          |
| AutoConnectClientLptPorts | ユーザーデバイス側のLPTポートへの自動接続                    | 1 (許可) または 0 (禁止) |

これらのレジストリを設定したら、そのマスターイメージまたは物理マシンが使用されるようにマシンカタログを変更します。ユーザーのデスクトップは、ログオフ時に新しい設定で更新されます。

# Connector for Configuration Manager 2012のポリシー設定

Jan 04, 2017

[Connector for Configuration Manager 2012] カテゴリには、Citrix Connector 7.5エージェントを構成するための設定項目が含まれています。

**重要：**警告、ログオフ、および再起動メッセージに関する設定項目は、手動管理またはProvisioning Servicesで管理するサーバーOSマシンカタログにのみ適用されます。これらのマシンカタログでは、保留中のアプリケーションのインストールまたはソフトウェアのアップデートがある場合、Connectorサービスによりユーザーに警告が表示されます。

MCSで管理するカタログでは、Studioでユーザーに通知してください。手動管理のデスクトップOSカタログでは、Configuration Managerでユーザーに通知してください。Provisioning Servicesで管理するデスクトップOSカタログでは、Provisioning Servicesでユーザーに通知してください。

## 警告表示間隔

この設定項目では、警告メッセージを表示する間隔を定義します。

間隔は、<ddd>.<hh>:<mm>:<ss>形式で設定します。

- <ddd>はオプションのパラメーターで、日数を0~999で指定します。
- <hh>は時間で、0~23で指定します。
- <mm>は分で、0~59で指定します。
- <ss>は秒で、0~59で指定します。

デフォルトでは、01:00:00 (1時間) が設定されています。

## 警告メッセージの内容

この設定項目では、予定されているソフトウェア更新、またはログオフが必要となるメンテナンスをユーザーに通知するためのメッセージを入力します。

デフォルトでは、「{TIMESTAMP} Please save your work.The server will go offline for maintenance in {TIMELEFT}」というメッセージが設定されています。

## 警告メッセージのタイトル

この設定項目では、警告メッセージのタイトルバーに表示される文字列を入力します。

デフォルトでは、「Upcoming Maintenance」というタイトルが設定されています。

## 警告表示期間

この設定項目では、ソフトウェアの更新またはメンテナンスについての警告メッセージを表示する期間を定義します。

期間は、<ddd>.<hh>:<mm>:<ss>形式で設定します。

- <ddd>はオプションのパラメーターで、日数を0~999で指定します。
- <hh>は時間で、0~23で指定します。
- <mm>は分で、0~59で指定します。
- <ss>は秒で、0~59で指定します。

デフォルトでは、16:00:00 (16時間) が設定されています。これにより、メンテナンスの約16時間前に最初の警告メッセージ

が表示されます。

### 最終的な強制ログオフメッセージの内容

この設定項目では、強制ログオフ処理が開始されたことをユーザーに通知するためのメッセージを入力します。

デフォルトでは、「The server is currently going offline for maintenance」というメッセージが設定されています。

### 最終的な強制ログオフメッセージのタイトル

この設定項目では、最終的な強制ログオフメッセージのタイトルバーに表示される文字列を入力します。

デフォルトでは、「Notification From IT Staff」というタイトルが設定されています。

### 強制ログオフの猶予期間

この設定項目では、ソフトウェアの更新またはメンテナンスのために、ユーザーにログオフを警告してから実際に強制ログオフ処理を開始するまでの待機期間を定義します。

期間は、<ddd>.<hh>:<mm>:<ss>形式で設定します。

- <ddd>はオプションのパラメーターで、日数を0~999で指定します。
- <hh>は時間で、0~23で指定します。
- <mm>は分で、0~59で指定します。
- <ss>は秒で、0~59で指定します。

デフォルトでは、00:05:00 (5分) が設定されています。

### 強制ログオフメッセージの内容

この設定項目では、強制ログオフが開始される前に作業を保存してログオフするようにユーザーに通知するためのメッセージを入力します。

デフォルトでは、「{TIMESTAMP} Please save your work and log off.The server will go offline for maintenance in {TIMELEFT}」というメッセージが設定されています。

### 強制ログオフメッセージのタイトル

この設定項目では、強制ログオフメッセージのタイトルバーに表示される文字列を入力します。

デフォルトでは、「Notification From IT Staff」というタイトルが設定されています。

### イメージ管理モード

Connectorエージェントでは、Provisioning ServicesまたはMCSで管理されるマシンのクローン上で動作しているかどうか自動的に検出されます。これらのイメージ管理されたクローン上では、Configuration ManagerによるアップデートがConnectorエージェントによってブロックされ、カタログのマスターイメージ上にアップデートが自動的にインストールされます。

マスターイメージのアップデートが完了したら、StudioでMCSカタログクローンの再起動をオーケストレーションします。Connectorエージェントは、Configuration ManagerのメンテナンスウィンドウでPVSカタログクローンの再起動を自動的にオーケストレーションします。この動作を無効にして、Configuration Managerによってソフトウェアがカタログクローンにインストールされるように設定するには、イメージ管理モードを [無効] に変更します。

### 再起動メッセージの内容

この設定項目では、サーバーの再起動をユーザーに通知するためのメッセージを入力します。

デフォルトでは、「The server is currently going offline for maintenance」というメッセージが設定されています。

### 定期的なエージェントタスクの実行間隔

この設定項目では、Citrix Connectorエージェントタスクの実行間隔を指定します。

期間は、<ddd>.<hh>:<mm>:<ss>形式で設定します。

- <ddd>はオプションのパラメーターで、日数を0~999で指定します。
- <hh>は時間で、0~23で指定します。
- <mm>は分で、0~59で指定します。
- <ss>は秒で、0~59で指定します。

デフォルトでは、00:05:00（5分）が設定されています。



# 印刷

Jan 04, 2017

環境でのプリンター管理には、以下の複数の段階があります。

1. 印刷の概念を理解します。
2. 印刷アーキテクチャを計画します。これには、業務上のニーズや既存の印刷インフラストラクチャについての分析と、ユーザーやアプリケーションが現状でどのように印刷を行っているか、および理想的な印刷管理モデルは何かについての評価が含まれます。
3. プリンタープロビジョニングの方法を選択し、印刷設計を展開するためのポリシーを作成して印刷環境を構成します。新しい従業員またはサーバーが追加されたときにポリシーを更新します。
4. 新しい印刷環境を実務環境に展開する前に、その環境をテストします。
5. プリンタードライバーを管理し、印刷のパフォーマンスを最適化してCitrixの印刷環境を維持します。
6. 発生する問題をトラブルシューティングします。

## 印刷の概念

印刷環境の構築を計画する前に、Citrix環境での印刷処理の主な概念について理解しておく必要があります。

- 使用できるプリンタープロビジョニングの種類
- 印刷ジョブをどのようにルーティングするか
- プリンタードライバーの基本的な管理方法

印刷の概念は、Windowsの印刷概念上に構成されています。環境での印刷設定を正しく管理するには、Windowsでのネットワークやクライアント印刷のしくみについて熟知しており、それが実際の環境にどのように適用されるのかを理解する必要があります。

## 印刷プロセス

この環境では、ユーザーによる印刷はすべてアプリケーションをホストするマシン上で開始されます。印刷ジョブはネットワークプリントサーバーまたはユーザーデバイスを介して印刷装置にリダイレクトされます。

仮想デスクトップやアプリケーションのユーザーに提供されるワークスペースは永続的ではありません。ユーザーのセッションが終了すると、そのユーザーのワークスペースはサーバーから削除されます。このため、各セッションの開始時にすべての設定を再構築する必要があります。この結果、ユーザーが新しいセッションを開始するたびに、ユーザーのワークスペースが再構築されます。

ユーザーが印刷を実行すると、以下の処理が行われます。

- ユーザーに提供されるプリンターを決定します。この処理は、プリンタープロビジョニングと呼ばれます。
- ユーザーの印刷設定を復元します。
- セッションのデフォルトプリンターを決定します。

管理者は、プリンタープロビジョニング、印刷ジョブの送信経路、プリンタープロパティの保存、およびプリンタードライバー管理に関するオプションを変更して、上記の処理をカスタマイズできます。これらのオプションの変更によって環境での印刷パフォーマンスやユーザーエクスペリエンスがどのように変化するかを検証してください。

## プリンタープロビジョニング

セッション用のプリンターを準備する処理は、プリンタープロビジョニングと呼ばれます。通常、この処理は動的に行われます。つまり、セッションで提供されるプリンターは事前定義されておらず、非永続的です。プリンターは、セッションへのログオン時または再接続時にポリシーに基づいて構成されます。このため、ポリシー、ユーザーの場所、およびネットワ

クに基づいて、異なるプリンターをユーザーに提供できます。つまり、ユーザーが別の場所に移動すると、そのユーザーの印刷環境が変更されます。

このCitrix製品の環境では、クライアント側のプリンターが監視され、クライアント側プリンターの追加、削除、および変更に応じてセッションの自動作成プリンターが動的に変更されます。この動的プリンター検出は、さまざまなデバイスを使用するモバイルユーザーにとって便利な機能です。

プリンターのプロビジョニングには、主に以下の方法があります。

- **Universal Print Server** — Citrix [Universal Print Server](#)は、ネットワークプリンターでのユニバーサル印刷をサポートします。Universal Print Serverでは、ユニバーサルプリンタードライバが使用されます。これにより、サーバーOSマシン上の単一のドライバを使って、任意のデバイスからネットワーク印刷を実行できます。リモートの印刷サーバーを使う環境では、Citrix Universal Print Serverの使用をお勧めします。Universal Print Serverで送信される印刷ジョブは最適化および圧縮されるため、ネットワーク消費を抑えてユーザーエクスペリエンスを向上させることができます。

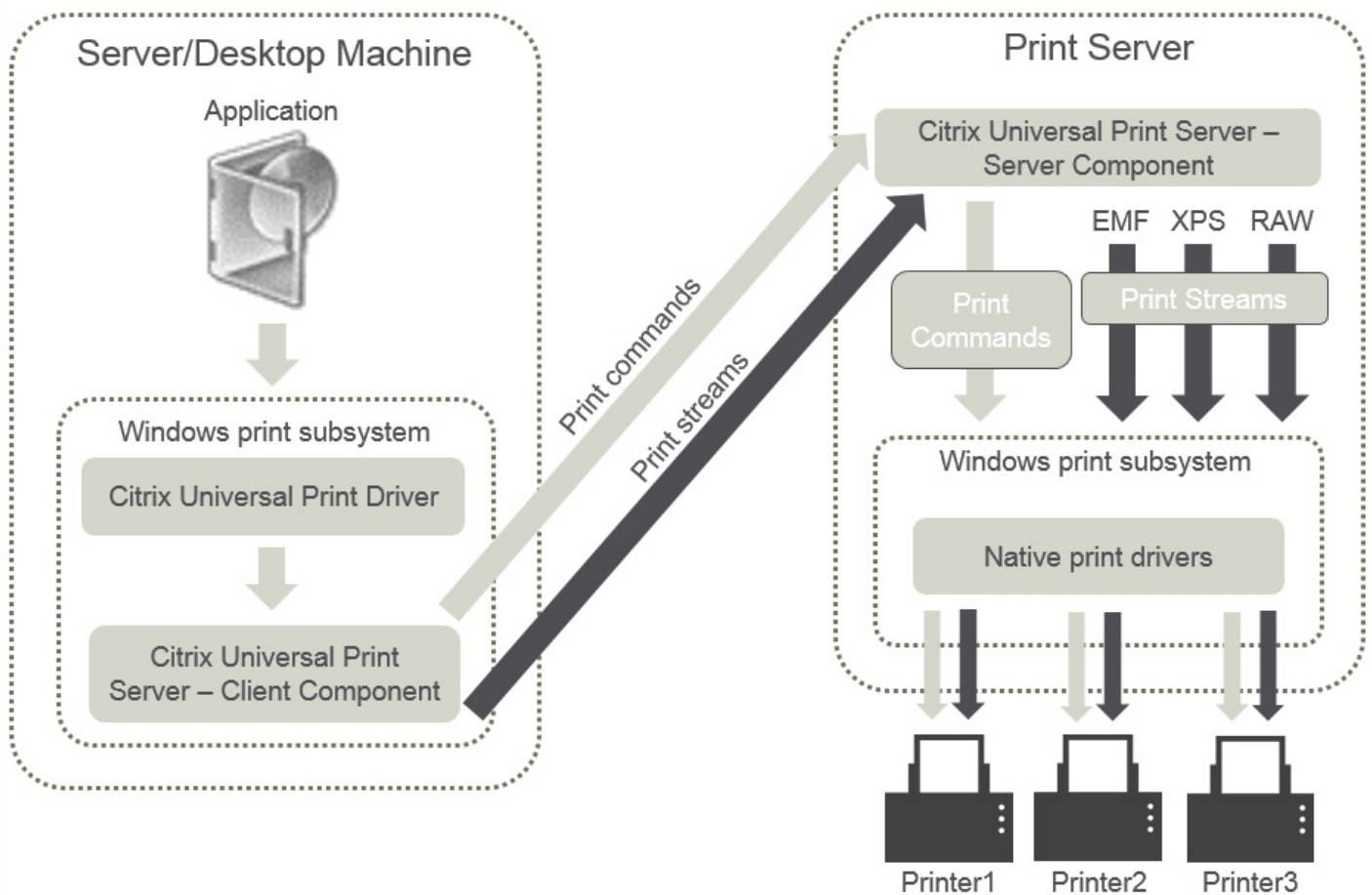
Universal Print Serverの機能は以下のコンポーネントで構成されます。

- クライアントコンポーネントのUPClient — サーバーOSマシンでユニバーサルプリンタードライバを使用するセッションネットワークプリンターをプロビジョニングする場合は、そのマシン上のUPClientを有効にします。
- サーバーコンポーネントのUPServer — プリントサーバーでユニバーサルプリンタードライバを使用するセッションネットワークプリンターをプロビジョニングする場合は（それらのセッションプリンターを集中管理的にプロビジョニングするかどうかに関係なく）、その印刷サーバー上にUPServerをインストールします。

Universal Print Serverの要件とセットアップ詳細については、システム要件およびインストールに関するドキュメントを参照してください。

注：Universal Print ServerはVDI-in-a-Box 5.3でもサポートされます。VDI-in-a-BoxでのUniversal Print Serverのインストールについて詳しくは、VDI-in-a-Boxのドキュメントを参照してください。

次の図は、Universal Print Serverを使用する環境におけるネットワークベースのプリンターの一般的なワークフローを示しています。



Citrix Universal Print Serverを有効にすると、接続されているすべてのネットワークプリンターでUniversal Print Serverが自動検出されて使用されます。

- **自動作成 - 自動作成**とは、各セッションの開始時に自動的に作成されるプリンターを指します。リモートネットワークプリンターとローカルに接続されたクライアントプリンターの両方を自動作成できます。ユーザーあたりのプリンター数が多い環境では、デフォルトのクライアントプリンターだけが自動作成されるように構成することを検討します。自動作成するプリンターの数をもっと少なくすることで、サーバーOSマシンの負荷（メモリやCPU）を軽減できます。また、これによりユーザーログオン時間も短縮されます。

以下の項目に基づいてプリンターが自動作成されます。

- ユーザーデバイス上にインストールされたプリンター。
- セッションに適用されるポリシー。  
管理者は、自動作成に関するポリシーを設定して、作成されるプリンターの数や種類を制御できます。デフォルトでは、ユーザーデバイス上で設定されているすべてのプリンター（ローカル接続のプリンターおよびネットワークプリンター）が自動作成され、ユーザーに提供されます。

ユーザーがセッションを終了すると、これらのプリンターは削除されます。

クライアントプリンターおよびネットワークプリンターの自動作成機能を使用する場合、保守作業が必要です。たとえば、プリンターを追加した場合は以下の設定が必要になります。

- [セッションプリンター] ポリシー設定を更新します。
- [プリンタードライバーのマッピングと互換性] ポリシー設定ですべてのサーバーOSマシンにドライバーを追加しま

す。

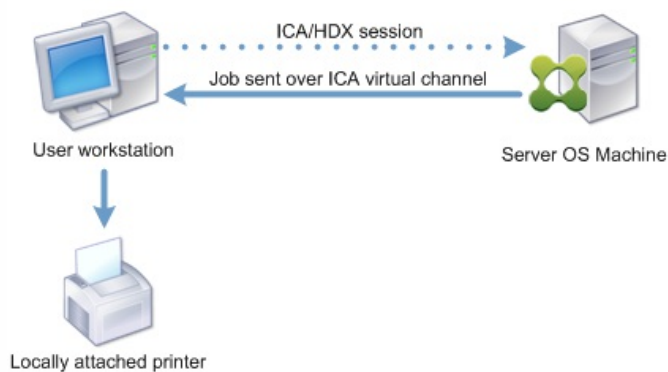
## 印刷ジョブの送信

印刷経路という語は、印刷ジョブがどのような経路で印刷装置に転送されるのか、および印刷ジョブがどこにスプールされるのかという概念を含んでいます。印刷環境を管理する場合、これらの概念を理解することは重要です。印刷ジョブのルーティング経路はネットワークトラフィックに影響し、スプール場所は印刷ジョブを処理するコンピューターの負荷に影響します。

この環境では、印刷装置への印刷ジョブの転送経路として、クライアント経由とネットワーク上のプリントサーバー経由の2つがあります。これらの転送経路は、「クライアント印刷経路」および「ネットワーク印刷経路」と呼ばれます。デフォルトでどちらの印刷経路が使用されるかは、使用されるプリンターの種類により異なります。

### ローカル接続のプリンター

印刷ジョブは、サーバーOSマシンからクライアントに送信され、さらにローカル接続のプリンターに転送されます。この場合、ICAプロトコルにより最適化および圧縮された印刷ジョブがネットワーク上に送信されます。印刷装置がユーザーデバイスにローカルに接続されている場合、印刷ジョブがICA仮想チャネルで転送されます。



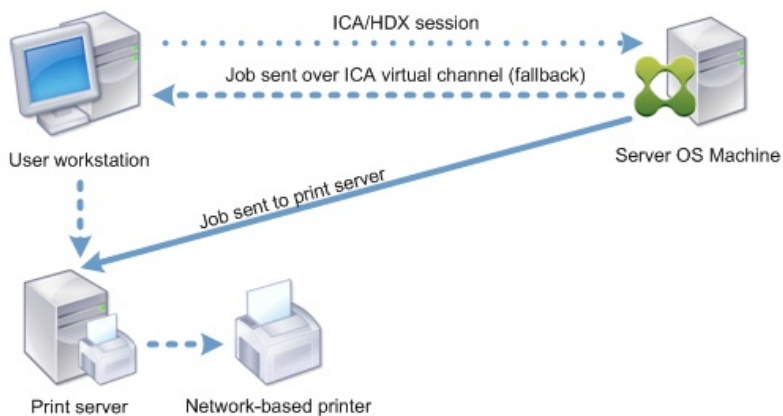
### ネットワーク上のプリンター

デフォルトでは、サーバーOSマシンからのすべての印刷ジョブがネットワークを介してプリントサーバーに直接転送されます。ただし、以下の状況では印刷ジョブが自動的にICA仮想チャネルで転送されます。

- 仮想デスクトップまたはアプリケーションがプリントサーバーにアクセスできない場合。
- プリンター固有のドライバーがサーバーOSマシン上にない場合。

Universal Print Serverが無効な場合、ICA仮想チャネルを介して送信される印刷ジョブは最適化および圧縮されるため、WANなどの狭帯域幅接続で隔たれたサーバーとクライアント間でクライアント印刷経路が使用されるように構成するとネットワークトラフィックへの負担が軽減されます。

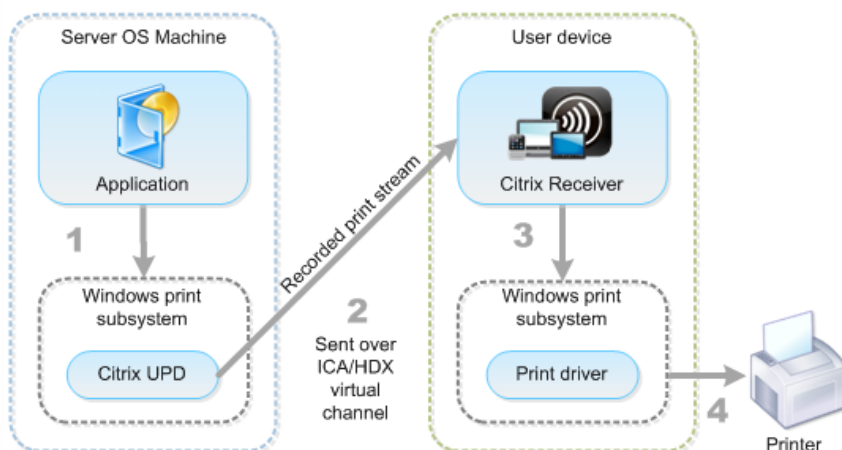
また、クライアント印刷経路では、印刷ジョブに割り当てられる帯域幅を制限できます。印刷機能がないシンクライアントなど、ユーザーデバイスを介して印刷ジョブを転送できない場合は、QoS設定でICA/HDXトラフィックを優先させて、セッションで良好なユーザーエクスペリエンスが提供されるように構成してください。



## プリンタードライバーの管理

この環境での印刷を簡素化するため、Citrixユニバーサルプリンタードライバーの使用をお勧めします。ユニバーサルプリンタードライバーは、あらゆる印刷デバイスをサポートする、デバイスに依存しないドライバーで、必要なドライバーの数を減らして管理業務を簡素化します。

次の図は、ユニバーサルプリンタードライバーコンポーネントとデバイスにローカル接続されたプリンターの一般的なワークフローを示しています。

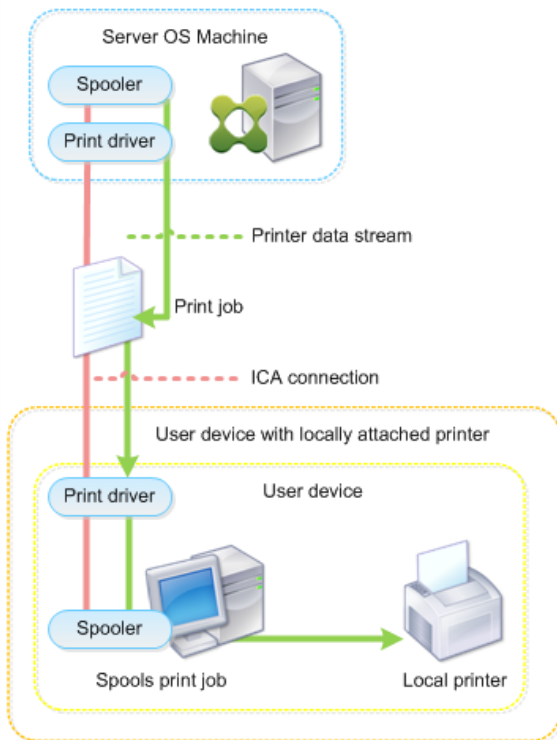


プリンタードライバーの管理方法を計画する場合、ユニバーサルプリンタードライバーを使用するか、デバイス固有のドライバーを使用するか、またはその両方を使用するかを決定する必要があります。標準ドライバーをサポートする場合は、以下の点を検討する必要があります。

- サポートするドライバーの種類。
- サーバーOSマシンにプリンタードライバーがない場合に自動的にインストールされるように設定するかどうか。
- プリンタードライバーの互換性リストを作成するかどうか。

プリンターの自動作成時に、ユーザーデバイスに接続された新しいローカルプリンターが検出されると、必要なプリンタードライバーについてサーバーOSマシンがチェックされます。デフォルトでは、Windowsネイティブドライバーが使用できない場合は、ユニバーサルプリンタードライバーが使用されます。

正しく印刷するには、サーバーOSマシン上のプリンタードライバーとユーザーデバイス上のドライバーが一致する必要があります。次の図は、クライアント印刷経路でサーバーとクライアント上のプリンタードライバーがどのように使用されるかを示しています。



#### 関連トピック

- 印刷構成の例
- ベストプラクティス、セキュリティに関する考慮事項、およびデフォルトの操作
- 印刷に関するポリシーと設定
- プリンターのプロビジョニング
- 印刷環境の保守
- Universal Print Serverの要件

# 印刷構成の例

Jan 04, 2017

組織のコンピューティング環境やユーザーのニーズに適した印刷環境を設定すると、管理が容易になります。通常、デフォルトの印刷構成でも正しく印刷できますが、ユーザーエクスペリエンスが低下したり、ネットワーク使用が最適化されなかったり、管理上のオーバーヘッドが生じたりする場合があります。

印刷環境を設定するときは、以下の事項を考慮します。

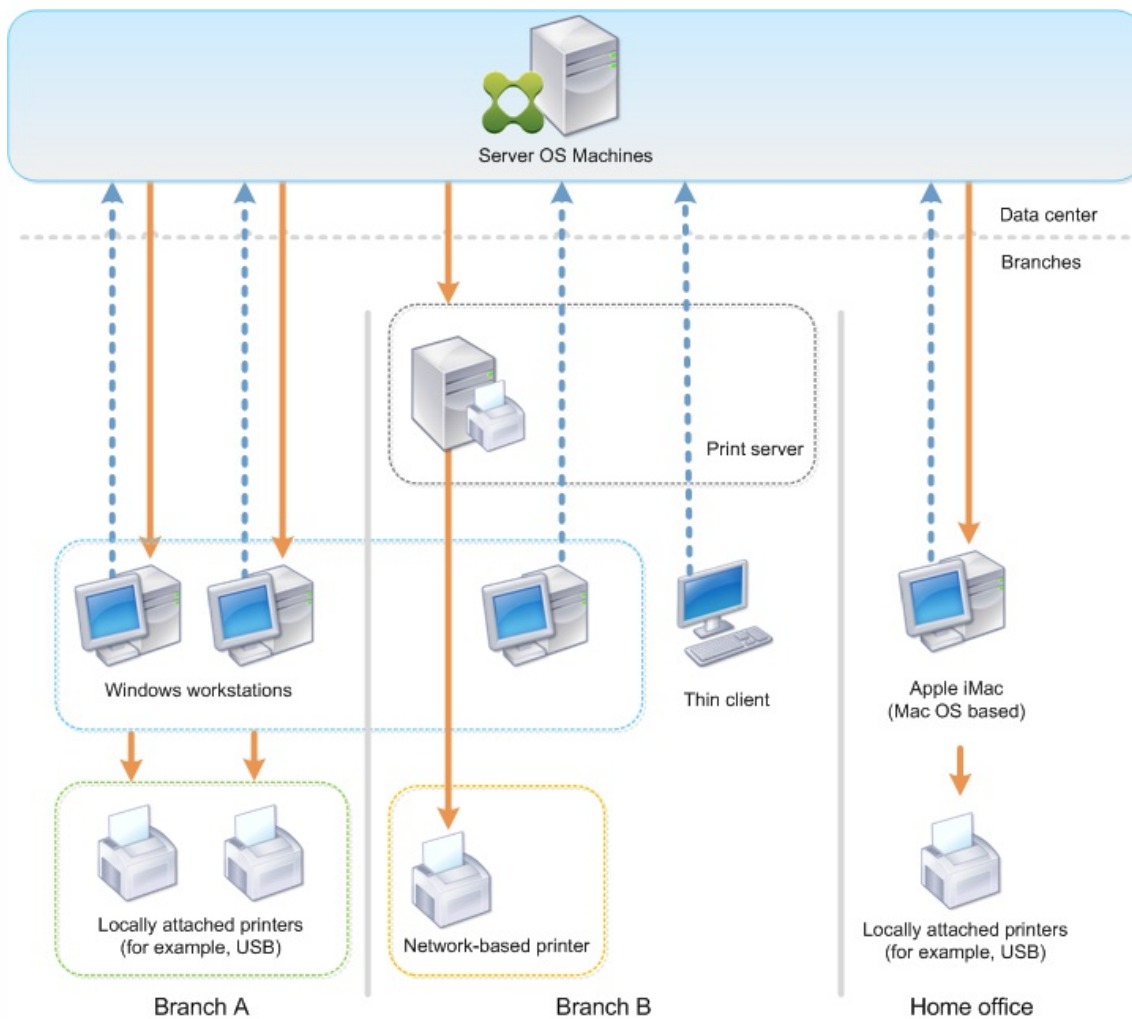
- 業務上のニーズと既存の印刷インフラストラクチャ。  
組織のニーズに基づいて、印刷環境を設計します。既存の印刷環境（ユーザーが自分でプリンターを追加できるかどうか、どのユーザーがどのプリンターにアクセスできるか、など）を確認し、それに沿って印刷環境を構成できます。
- 組織のセキュリティポリシー。人事部用のプリンターなど、特定のユーザー用に予約されたプリンターがあるかどうかを確認します。
- メインのワークステーションとは離れた場所で印刷するユーザーがいるかどうか。たとえば、複数のワークステーションを移動しながら作業したり、出張先で印刷したりするユーザーがいるかどうかを確認します。

印刷環境を設計するにあたり、ユーザーがローカルのユーザーデバイス上で印刷するときと同様なユーザーエクスペリエンスを提供することを目標にします。

## 印刷展開の例

次の図は、以下の環境での印刷展開を示しています。

- **支社A** — 数台のWindowsワークステーションがある海外の小さな支社。すべてのユーザーワークステーションは共有されていないプリンターにローカルで接続されています。
- **支店B** — シンクライアントおよびWindowsベースのワークステーションがある大規模な支社。効率を上げるため、この支店のユーザーはネットワーク上のプリンターを（各階で1台）共有しています。支店内に置かれているWindowsベースのプリントサーバーが印刷キューを管理します。
- **社員の自宅** — Mac OSベースのユーザーデバイスで自宅から会社のCitrixインフラストラクチャにアクセスしています。ユーザーデバイスはプリンターにローカルで接続されています。



以降のセクションでは、各印刷環境をシンプルにして簡単に管理するための構成について説明します。

### 自動作成されるクライアントプリンターとCitrixユニバーサルプリンタードライバー

支社AのすべてのユーザーはWindowsベースのワークステーションを使用しており、自動作成されたクライアントプリンターとユニバーサルプリンタードライバーが使用されます。この構成には以下のメリットがあります。

- パフォーマンス — 印刷ジョブはICA印刷チャネル上で配信されます。このため、印刷データの圧縮により帯域幅を節約できます。サイズの大きなドキュメントを印刷しているユーザーがほかのユーザーのセッションパフォーマンスを低下させることがないように、最大印刷帯域幅を指定するCitrixポリシーを構成します。

代替策として、マルチストリームICA接続を使用して、印刷トラフィックが優先度の低い専用のTCP接続で転送されるように構成することもできます。マルチストリームICAは、WAN接続にサービス品質 (QoS) が実装されていない場合のオプションです。

- 柔軟性 — Citrixユニバーサルプリンタードライバーを使用して、新しいプリンタードライバーをデータセンターに追加することなく、クライアントに接続されているすべてのプリンターを仮想デスクトップやアプリケーションのセッションでも使用できるように構成します。

### Citrix Universal Print Server

支社Bのすべてのプリンターはネットワークに接続されており、その印刷キューはWindowsプリントサーバー上で管理されます。このため、Citrix Universal Print Serverが最も効果的な構成になります。



必要なすべてのプリンタードライバーは、ローカルの管理者によりプリントサーバー上にインストールされて管理されます。ネットワーク上のプリンターは、以下のように仮想デスクトップやアプリケーションセッションにマップされます。

- Windowsベースのワークステーションの場合 — ローカルのITチームの支援により、ユーザーのWindowsワークステーションを適切なネットワークプリンターに接続します。これにより、ユーザーはローカルにインストールされたアプリケーションから印刷できるようになります。

仮想デスクトップやアプリケーションのセッションを開始すると、ローカルで構成されたプリンターがセッション内で自動作成されます。仮想デスクトップまたはアプリケーションは、可能であれば直接ネットワーク接続としてプリントサーバーに接続します。

Citrix Universal Print Serverコンポーネントが構成されているため、ネイティブのプリンタードライバーは不要です。ドライバーをアップデートしたりプリンターキューを変更したりしても、データセンターで何らかの構成を行う必要はありません。

- シンククライアントの場合 — シンククライアントデバイスのユーザーは、仮想デスクトップやアプリケーションのセッション内でプリンターを接続する必要があります。ユーザーに最もシンプルな印刷構成を提供するには、管理者がCitrixポリシーの [セッションプリンター] 設定を階ごとに構成して、各階のプリンターがデフォルトのプリンターとして接続されるようにします。

ユーザーが階を移動しても正しいプリンターが接続されるようにするには、シンククライアントのサブセットまたは名前についてポリシーが適用されるように構成します。この構成は「近接プリンター機能」と呼ばれ、ローカルプリンタードライバーのメンテナンスを委任管理モデルに基づいて実行できます。

プリンターキューを変更または追加する必要がある場合は、Citrix管理者が適切なポリシーの [セッションプリンター] 設定を変更する必要があります。

ネットワーク印刷トラフィックはICA仮想チャネルの外側で送信されるため、サービス品質 (QoS) が実装されます。ICA/HDXトラフィックにより使用されるポート上の送受信ネットワークトラフィックは、ほかのすべてのネットワークトラフィックよりも優先されます。この構成により、大きな印刷ジョブがユーザーセッションに影響を及ぼすことがなくなります。

## 自動作成されるクライアントプリンターとCitrixユニバーサルプリンタードライバー

ユーザーが自宅で非標準的なワークステーションを使用し、管理されていない印刷装置を使用する場合、ユニバーサルプリンタードライバーを使用してクライアントプリンターを自動作成する構成が最適です。

## 展開の要約

要約すると、展開例は以下のように構成されています。

- サーバーOSマシン上にはプリンタードライバーがインストールされていません。Citrixユニバーサルプリンタードライバーのみを使用します。ネイティブのプリンタードライバーへのフォールバックおよびプリンタードライバーの自動インストールは無効です。
- すべてのクライアントプリンターを自動作成するためのポリシーがすべてのユーザーに適用されます。サーバーOSマシンはデフォルトでプリントサーバーに直接アクセスします。必要な構成タスクは、Universal Print Serverコンポーネントの有効化のみです。
- 支社Bの各階に個別の [セッションプリンター] 設定が構成されており、その階のすべてのシンククライアントに適用されます。
- 支社BにはQoSが実装され、優れたユーザーエクスペリエンスが提供されています。

# ベストプラクティス、セキュリティに関する考慮事項、およびデフォルトの操作

Jan 04, 2017

## ベストプラクティス

環境での最適な印刷ソリューションは、さまざまな要因により決定されます。以下のベストプラクティスの中には、特定のサイトに適用されない場合があります。

- Citrix Universal Print Serverを使用します。
- ユニバーサルプリンタードライバーまたはWindowsネイティブドライバーを使用します。
- サーバーOSマシン上にインストールされるプリンタードライバー数を最小化します。
- ネイティブドライバーへのドライバーマッピングを使用します。
- 動作検証されていないプリンタードライバーを実稼働環境サイトにインストールしないようにします。
- ドライバーのアップデートインストールを避け、常にドライバーをアンインストールしてからプリントサーバーを再起動して、その後で新しいドライバーをインストールしてください。
- 未使用のドライバーをアンインストールするか、[プリンタードライバーのマッピングと互換性] ポリシーを行使して、プリンターがそのドライバーで作成されないようにします。
- Version 2のカーネルモードドライバーを使用しないようにします。
- 特定のプリンターがサポートされるかどうかについては、製造元に問い合わせるか、Citrix Ready製品に関する情報 ([www.citrix.com/ready](http://www.citrix.com/ready)) を参照してください。

一般的に、Microsoft社より提供されるプリンタードライバーはすべてTerminal Servicesでテストされ、Citrix環境での動作が確認されています。ただし、サードパーティ製のプリンタードライバーを使う前に、ターミナルサービスでの動作がWindows Hardware Quality Labs (WHQL) プログラムで認定されているかどうかをプリンタードライバーのベンダーに確認してください。Citrixではプリンタードライバーの動作を保証しません。

## セキュリティに関する注意事項

Citrixの印刷ソリューションは、これ自体がセキュアに設計されています。

- Citrix Print Manager Serviceは、ログオンやログオフ、切断、再接続、およびセッション終了などのセッションイベントを常に監視してそれらに応答します。実際のセッションユーザーを偽装して、サービス要求を処理します。
- Citrixの印刷ソリューションでは、セッション内の一意な名前空間に各プリンターが割り当てられます。
- Citrixの印刷ソリューションでは、自動作成プリンターにデフォルトのセキュリティ記述子が設定されます。これにより、あるセッションで自動作成されたクライアントプリンターにほかのセッションのユーザーがアクセスできないようになります。デフォルトでは、クライアントプリンターのアクセス権を変更するための管理者権限を持つユーザーでも、ほかのセッションのクライアントプリンターに誤って出力してしまうということはありません。

## デフォルトの印刷動作

印刷に関するポリシーを設定しない場合、デフォルトで次のように処理されます。

- Universal Print Serverが無効になります。
- ユーザーデバイス上で設定されているすべてのプリンターが、各セッションの開始時にサーバー上に自動作成されます。この動作は、Citrixポリシーの[クライアントプリンターを自動作成する]設定で[すべてのクライアントプリンターを自動作成する]を構成した場合と同等です。
- クライアントデバイスにローカル接続されたプリンターへのすべての印刷ジョブは、ICAチャネルを介してユーザーデバイスに送信され、プリンターに転送されます(クライアント印刷経路)。
- ネットワークプリンターへのすべての印刷ジョブは、サーバーOSマシンからプリントサーバーに直送されます。印刷ジョブ

ブをネットワーク上に送信できない場合は、ユーザーコンピューターを介して転送されます（リダイレクトされるクライアント印刷ジョブ）。

この動作は、Citrixポリシーの [プリントサーバーへの直接接続] 設定で [無効] を選択した場合と同等です。

- デフォルトでは、印刷プロパティ（ユーザーの印刷設定とデバイス設定）はユーザーデバイス上に格納されます。クライアント側でこの処理がサポートされない場合、サーバーOSマシン上のユーザープロファイルに印刷プロパティが格納されます。

この動作は、Citrixポリシーの [プリンタープロパティの保存] 設定で [クライアントに保存できない場合にのみユーザープロファイルに保存する] を選択した場合と同等です。

- セッション内でプリンターが自動作成されるときに、そのサーバーOSマシン上にインストールされているWindowsバージョンのプリンタードライバーが使用されます。適切なドライバーがインストールされていない場合、Windowsオペレーティングシステムからドライバーがインストールされます。Windowsオペレーティングシステムから適切なドライバーをインストールできない場合、Citrixユニバーサルプリンタードライバーが使用されます。

この動作は、Citrixポリシーの [付属のプリンタードライバーの自動インストール] 設定で [有効] を選択し、[ユニバーサル印刷] 設定で [要求されたドライバーを使用できない場合にのみユニバーサル印刷を使用する] を選択した場合と同等です。

ただし、[付属のプリンタードライバーの自動インストール] を有効にすると、必要以上に多くのプリンタードライバーがインストールされる可能性があります。

注：印刷に関するデフォルト設定を確認するには、新しいCitrixポリシーを作成し、印刷に関するすべての設定項目で [デフォルト値を使用する] チェックボックスをオンにします。これにより、デフォルトの設定が適用されます。

#### Always-Onログ

XenAppおよびXenDesktop 7.6 FP3にはプリントサーバーおよびVDAの出力サブシステムのためのAlways-Onログ機能があります。

ログをZIPとしてまとめてメールで送信するため、あるいは自動的にCitrix Insights Servicesにアップロードするには、7.6 FP3のVDAインストーラーで提供されるPowerShellコマンドレット (Start-TelemetryUpload) を使用します。

# 印刷に関するポリシーと設定

Jan 04, 2017

Citrixポリシーでは、ユーザーが公開アプリケーションからプリンターにアクセスするときの以下の動作を制御できます。

- どのようにプリンターを提供するか（どのようにセッションに追加するか）
- 印刷ジョブをどのようにルーティングするか
- プリンタードライバーをどのように管理するか

Citrixポリシーでは、ユーザーが使用するユーザーデバイスやユーザーアカウントなどの条件に応じて、異なる印刷環境を構築できます。

印刷機能の多くは、Citrixポリシーの [印刷] カテゴリにある以下の設定項目で制御できます。 [印刷] カテゴリの設定項目は、Citrixポリシーの標準的な動作に基づいて適用されます。

プリンター設定は、セッション終了時にプリンターオブジェクトまたは（ユーザーのネットワークアカウントに適切な権限がある場合は）クライアントの印刷装置に格納されます。Receiverのデフォルトでは、プリンターオブジェクトに格納された設定がまずチェックされ、見つからない場合はほかの場所に格納されている設定が使用されます。

デフォルトでは、ユーザーデバイス（デバイスがこれをサポートする場合）またはサーバーOSマシン上のユーザープロファイルにプリンターのプロパティが格納（または保持）されます。セッションでの作業中にユーザーがプリンターのプロパティを変更すると、その内容はそのマシン上のユーザープロファイルに反映されます。ユーザーがそのマシンに再ログオンしたり再接続したりすると、ユーザープロファイルに保持されたプロパティがユーザーデバイスに継承されます。つまり、ユーザーデバイス上のプリンタープロパティの変更は、ユーザーの次回ログオン時まで反映されません。

## 印刷設定の場所

Windowsの印刷環境では、印刷設定に対する変更をローカルコンピューターに格納したり、ドキュメントファイルに格納したりできます。この環境では、ユーザーが変更した印刷設定を以下の場所に格納できます。

- **ユーザーデバイス上**：Windowsユーザーは、ユーザーデバイス側の印刷設定を自分で変更できます。これを行うには、コントロールパネルでプリンターを右クリックして、[印刷設定] を選択します。たとえば、印刷の方向として [横] を選択すると、そのプリンターのデフォルトの方向として横向きが設定されます。
- **ドキュメント内**：ワードプロセッサやデスクトップパブリッシングのプログラムでは、印刷の向きなどのドキュメント設定はそのドキュメントファイル内に格納されます。たとえば、Microsoft Wordドキュメントを印刷キューに送ると、ユーザーが指定した印刷の向きやプリンター名などの印刷設定がそのドキュメントファイル内に格納されます。これらのオプションは、次回そのドキュメントを印刷するときのデフォルト設定として表示されます。
- **セッションでのユーザーによる変更**：自動作成されたプリンターでは、ユーザーがセッション内のコントロールパネルで変更したオプション、つまりサーバーOSマシン上で変更されたオプションだけが保持されます。
- **サーバーOSマシン上**：サーバーOSマシン上の特定のプリンタードライバーに対するデフォルト設定は、そのマシン上に格納されます。

Windowsベースの環境で保持される設定は、ユーザーがどのようにその設定を変更したかにより異なります。つまり、スプレッドシートプログラムなどに表示される印刷設定が、ドキュメントなどほかの場所に格納されている設定と異なることがあります。この結果、特定のプリンターに適用される設定は、セッション内で変化することがあります。

## ユーザーの印刷設定の階層構造

印刷に関するユーザー設定はさまざまな場所に格納されるため、特定の優先順位でそれらの設定が処理されます。また、デバイス設定はドキュメント設定とは区別され、より優先されることに注意してください。

デフォルトでは、ユーザーがセッション内で変更したすべての印刷設定、つまり保持された設定が適用され、その後でそのに

かの設定がチェックされます。ユーザーが印刷を行うと、サーバーOSマシン上に格納されたデフォルトの設定と、保持された設定やクライアントプリンター設定が統合されます。

## ユーザーの印刷設定の保存

プリンタープロパティの格納場所を変更することは推奨されません。デフォルトの格納場所（つまりユーザーデバイス上）を使用すると、ユーザーの印刷に一貫したプロパティが適用されるようになります。ユーザーデバイス上にプロパティを保存できない場合は、自動的にサーバーOSマシン上のユーザープロファイルが格納場所として使用されます。

以下の環境では、[プリンタープロパティの保存] 設定の内容を確認してください。

- ユーザーデバイス上へのプリンタープロパティの格納をサポートしない従来のプラグインソフトウェアが使用されている。
- 固定プロファイルを使用するWindowsネットワーク環境で、ユーザーのプリンタープロパティが保持されるように設定する。

# プリンターのプロビジョニング

Jan 04, 2017

プリンターのプロビジョニングには以下の3つの方法があります。

- [Citrix Universal Print Server](#)
- [クライアントプリンターの自動作成](#)
- [ユーザーへのネットワークプリンターの割り当て](#)

## Citrix Universal Print Server

環境に最適の印刷ソリューションを決定するときは、以下の点について検討します。

- Universal Print Serverにより提供されるイメージとフォントのキャッシュ、高度圧縮、最適化、QoSサポートなどの機能は、Windowsの印刷プロバイダーでは提供されません。
- ユニバーサルプリンタードライバでは、Microsoftによって定義されているパブリックな非デバイス依存の設定がサポートされます。ユーザーがプリンターの製造元固有のデバイス設定を使用する必要がある場合は、Universal Print ServerとWindowsネイティブドライバの両方を提供します。この構成では、Universal Print Serverの長所を維持したままでユーザーに特殊なプリンター機能へのアクセスが提供されます。ここで考慮すべきことは、Windowsネイティブドライバではメンテナンスが必要になるということです。
- Citrix Universal Print Serverは、ネットワークプリンターでのユニバーサル印刷をサポートします。Universal Print Serverではユニバーサルプリンタードライバが使用されます。このドライバはサーバーOSマシン上の単一のドライバで、シンクライアントやタブレットを含むあらゆるデバイスからのローカル印刷またはネットワーク印刷が可能になります。

Universal Print ServerをWindowsネイティブドライバと一緒に使うには、Universal Print Serverを有効にします。デフォルトでは、Windowsネイティブドライバが使用可能な場合はそれが使用されます。使用できない場合は、ユニバーサルプリンタードライバが使用されます。Windowsネイティブドライバのみ、またはユニバーサルプリンタードライバのみを使用するなど、ユニバーサル印刷機能の動作を変更するには、ポリシーの [ユニバーサル印刷の使用] 設定を使用します。

## Citrix Universal Print Server (UPS) のインストール

注：

- Universal Print Serverの32ビットオペレーティングシステム上での使用はサポートされていません。
- 印刷操作の間にユーザー認証を実行するには、Universal Print ServerがリモートデスクトップサービスVDAと同じドメインに参加する必要があります。
- UPClientコンポーネントは、VDAと一緒にインストールされます。クライアントコンポーネントを手動でインストールする必要はありません。

Citrix Universal Print Serverをインストールするには

1. Windows 2008 R2 SP1、Windows Server 2012、またはWindows Server 2012 R2プリントサーバー上のLTSRメタインストーラーで [拡張展開] > [Universal Print Server] をクリックし、画面の指示に従います。
2. サーバーを再起動します。

## クライアントプリンターの自動作成

ユニバーサル印刷ソリューションにより、クライアントプリンターに以下の機能が提供されます。

- **Citrixユニバーサルプリンター** -セッションの開始時に作成される汎用プリンターで、特定の印刷装置に関連付けられるものではありません。Citrixユニバーサルプリンターを使用することでログオン時のクライアントプリンターの列挙が不要に

なるため、リソース負荷が軽減され、ユーザーが高速にログオンできるようになります。ユニバーサルプリンターでは、クライアント側のあらゆる印刷装置を使用できます。

Citrixユニバーサルプリンターは、ユーザーが使用するユーザーデバイスやReceiverによっては正しく動作しない場合があります。CitrixユニバーサルプリンターはWindows環境で動作し、Citrix Offline Plug-inや、クライアント上にストリーム配信されるアプリケーションをサポートしません。このような環境では、クライアントプリンターの自動作成機能とユニバーサルプリンタードライバの使用を検討してください。

非Windows Receiverのユーザーにユニバーサル印刷ソリューションを提供するには、自動的にインストールされるPostScript/PCLベースのユニバーサルプリンタードライバを使用してください。

- Citrixユニバーサルプリンタードライバは、Citrixユニバーサルプリンタードライバを構成すると、デフォルトでEMFベースのユニバーサルプリンタードライバが使用されます。  
Citrixユニバーサルプリンタードライバによる印刷ジョブのサイズは、古いバージョンなどのプリンタードライバのものよりも小さい場合があります。ただし、特殊なプリンターでの印刷ジョブを最適化するには、デバイス固有のドライバが必要になる場合があります。

**ユニバーサル印刷の構成** — 以下のCitrixポリシー設定を使用してユニバーサル印刷を構成します。詳しくは、画面に表示される各ポリシー設定のヘルプを参照してください。

- ユニバーサル印刷の使用：ユニバーサル印刷を使用する条件を指定します。
- 汎用ユニバーサルプリンターを自動作成する：ユニバーサル印刷と互換性があるユーザーデバイスが使用されたセッションで、汎用的なCitrixユニバーサルプリンターオブジェクトの自動作成を有効または無効にします。デフォルトでは、汎用ユニバーサルプリンターオブジェクトは自動作成されません。
- ユニバーサルドライバの優先度：ユニバーサルプリンタードライバの使用優先順位を指定します。一覧の上位にあるドライバから順に使用されます。この一覧では、ドライバを追加、編集、または削除したり、優先順位を変更したりできます。
- ユニバーサル印刷プレビューの設定：自動作成プリンターおよび汎用ユニバーサルプリンターの印刷プレビュー機能を使用するかどうかを指定します。
- ユニバーサル印刷EMF処理モード：Windowsユーザーデバイス上でのEMFスプールファイルの処理方法を制御します。デフォルトでは、EMFスプールファイルがクライアント上のスプールキューに直接挿入されます。これにより、EMF形式の印刷を高速に実行でき、CPUリソースの消費も少なくなります。

ポリシーについて詳しくは、「[印刷パフォーマンスの最適化](#)」を参照してください。用紙サイズ、印刷品質、色設定、両面印刷、部数などのデフォルト設定を変更する方法については、[CTX114420](#)を参照してください。

**ユーザーデバイスからのプリンターの自動作成** — デフォルトでは、セッションの開始時にユーザーデバイス上で設定されているすべてのプリンターが自動作成されます。管理者は、セッション内でユーザーに提供するプリンターの種類を制御して、自動作成を無効にできます。

自動作成機能を制御するには、Citrixポリシーの [クライアントプリンターを自動作成する] 設定を使用します。以下のオプションを選択できます。

- ローカル接続されているプリンターやネットワークプリンターを含め、ユーザーデバイス上で設定されているすべてのプリンターがセッション開始時に自動作成されるようにする（デフォルト）。
- ユーザーデバイスに物理的に接続されているすべてのローカルプリンターが自動作成されるようにする。
- ユーザーデバイス上で設定されているデフォルトプリンターだけが自動作成されるようにする。
- すべてのクライアントプリンターに対する自動作成を無効にする。

[クライアントプリンターを自動作成する] 設定を使用する場合は、[クライアントプリンターリダイレクト] 設定を [許可]（デフォルト）にする必要があります。

ユーザーへのネットワークプリンターの割り当て

デフォルトでは、クライアントデバイス上で設定されているすべてのネットワークプリンターが、セッション開始時に自動作成されます。管理者は、列挙およびマップされるプリンターの数を最小限にするために、各セッションで特定のネットワークプリンターだけが作成されるように構成することができます。このようなプリンターをセッションプリンターと呼びます。

IPアドレスによりセッションプリンターポリシーをフィルターして、近接プリンター機能を提供できます。この機能を使用すると、ユーザーのIPアドレスの範囲に応じて、特定のネットワークプリンターが自動的に割り当てられるようになります。近接プリンター機能はCitrix Universal Print Serverにより提供され、このセクションで説明する構成は必要ありません。

近接プリンター機能は、以下の環境で使用できます。

- 企業の社内ネットワークでユーザーのIPアドレスがDHCPサーバーにより自動的に割り当てられる。
- 組織内のすべての部署で、それぞれ異なるIPアドレス範囲が割り当てられる。
- 各部署のIPアドレス範囲内にネットワークプリンターが存在する。

近接プリンター機能を構成すると、従業員がある部署から別の部署に移動する場合でも追加の印刷装置の構成は必要ありません。移動先の部署のIPアドレス範囲でユーザーデバイスが認識されると、その範囲内のすべてのネットワークプリンターへのアクセスが可能になります。

**セッションで特定のプリンターがリダイレクトされるように構成する** — 管理者割り当てのプリンターを作成するには、Citrixポリシーの [セッションプリンター] 設定を構成します。この設定では、以下のいずれかの方法でネットワークプリンターを追加します。

- プリンターのUNCパスを \\<servername>\<printername>形式で入力します。
- ネットワーク上でプリンターの場所を参照します。
- 特定サーバー上のプリンターを参照します。サーバー名を \\<servername>形式で入力して [参照] をクリックします。

**重要：**特定のセッションに複数のポリシーが適用される場合、それらのポリシー（優先度の高いものから低いものまですべての）の [セッションプリンター] 設定で指定されているすべてのネットワークプリンターが自動作成されます。複数のポリシーにより同じプリンターの自動作成が適用される場合、最も優先度の高いポリシーの設定だけがそのプリンターのカスタムデフォルト設定として使用されます。

[セッションプリンター] 設定を使用すると、サブネットなどの条件により異なるポリシーが適用されるように構成して、ユーザーがセッションを開始した場所によって異なるネットワークプリンターが自動作成されるように制御できます。

**セッションのデフォルトネットワークプリンターを指定する** — デフォルトでは、ユーザーの現在のデフォルトプリンター（通常使うプリンター）がセッションのデフォルトプリンターとして使用されます。セッションのデフォルトのクライアントプリンターとして設定するプリンターを指定するには、Citrixポリシーの [デフォルトプリンター] 設定を構成します。

1. [デフォルトプリンター] 設定で、[デフォルトのクライアントプリンター] ボックスの一覧から、以下のいずれかのオプションを選択します。
  - ネットワークプリンター名。 [セッションプリンター] ポリシー設定で追加されたプリンターがこのメニューに表示されます。デフォルトプリンターとして指定するネットワークプリンターを選択します。
  - デフォルトプリンターの設定を変更しない。ターミナルサービスまたはWindowsのユーザープロファイルで設定されているデフォルトプリンターが使用されます。詳しくは、画面に表示される各ポリシー設定のヘルプを参照してください。
2. このポリシーの適用先として、ユーザーグループ（またはそのほかのフィルターオブジェクト）を指定します。

**近接プリンター機能を構成する** — Citrix Universal Print Serverでは、近接プリンター機能も提供されます。この場合、ここで説明されている構成は必要ありません。

1. 各サブネット（またはプリンターが設定されている場所）に応じて、異なるポリシーを作成します。
2. 各ポリシーの [セッションプリンター] 設定で、そのサブネットの場所に設置されているプリンターを追加します。
3. [デフォルトプリンター] 設定で、[デフォルトプリンターの設定を変更しない] を選択します。
4. 各ポリシーの適用先として、クライアントのIPアドレスを指定します。DHCP IPアドレス範囲が変更された場合は、これらのポリシーも更新する必要があります。





# 印刷環境の保守

Jan 04, 2017

印刷環境では、以下の保守作業を行います。

- プリンタードライバーを管理する。
- 印刷パフォーマンスを最適化する。
- プリンターを表示して印刷キューを管理する。

## プリンタードライバーの管理

管理上のオーバーヘッドや潜在的な問題を最小化するため、Citrixユニバーサルプリンタードライバーの使用をお勧めします。

自動作成に失敗すると、デフォルトで、Windowsで提供されているWindowsネイティブのプリンタードライバーがインストールされます。ドライバーが使用できない場合は、ユニバーサルプリンタードライバーが使用されます。プリンタードライバーのデフォルトについて詳しくは、「[ベストプラクティス、セキュリティに関する考慮事項、およびデフォルトの操作](#)」を参照してください。

Citrixユニバーサルプリンタードライバーが適さない環境では、サーバーOSマシン上にインストールするドライバーの数を少なくするためにプリンタードライバーをマップします。プリンタードライバーをマップすることで、以下のことが可能になります。

- 特定のプリンターでCitrixユニバーサルプリンタードライバーだけが使用されるようにする
- 特定のドライバーによるプリンターの作成を許可または禁止する
- 問題が生じるプリンタードライバーの代わりに正しく動作するプリンタードライバーを割り当てる
- クライアント側のプリンタードライバーの代わりにWindowsサーバー上で使用可能なドライバーを割り当てる

**プリンタードライバーの自動インストールを無効にする** —サーバーOSマシン間で一貫したプリンター構成を保つため、プリンタードライバーの自動インストールを無効にします。これはCitrixのポリシー、Microsoftのポリシー、またはその両方で設定できます。Windowsネイティブドライバーが自動的にインストールされないようにするには、Citrixポリシーの [付属のプリンタードライバーの自動インストール] 設定を無効にします。

**クライアントプリンタードライバーのマップ** —ユーザーがセッションにログオンするときに、プリンタードライバー名など、クライアント側のプリンターの情報が提供されます。クライアントプリンターの自動作成時に、クライアントから提供されたプリンターのモデル名に基づいて、Windowsサーバーのプリンタードライバーの名前が選択されます。次に、選択されたプリンタードライバーが自動作成プロセスで使用され、リダイレクトされるクライアント印刷キューが作成されます。

次の手順で、ドライバー置換規則を定義して、マップされたクライアントプリンタードライバーの印刷設定を編集します。

1. 自動作成クライアントプリンターのドライバー置換規則を指定するには、Citrixポリシーの[プリンタードライバーのマッピングと互換性] 設定を構成して、クライアント側のプリンタードライバーの名前を追加し、それに割り当てるサーバー側プリンタードライバーを指定します（[サーバー側プリンタードライバー] を選択して [ドライバーの検索] をクリック）。ここでは、ワイルドカード文字を使用できます。たとえば、すべてのHP社製プリンターで特定のドライバーを使用する場合は、「HP\*」と入力します。
2. プリンタードライバーの使用を禁止するには、ドライバー名を選択して[作成しない] を選択します。
3. 必要に応じて、既存のマッピングを編集したり、マッピングを削除したり、一覧のドライバーエントリの順位を変更したりできます。
4. マップされたクライアントプリンタードライバーの印刷設定を編集するには、[設定] をクリックして印刷品質、印刷の向き、印刷カラーなどの設定を指定します。プリンタードライバーでサポートされないオプションを選択した場合、そのオプションは無視されます。ここで選択するオプションは、ユーザーが前回のセッションで指定し、保持されていた設定よりも優先されます。
5. 一部のプリンター機能は特定のドライバーでのみ使用可能であるため、ドライバーをマップした後でプリンターの動作を

細にテストすることをお勧めします。

ユーザーがログオンすると、クライアントプリンタードライバーの互換性一覧がチェックされ、その後でクライアントプリンターがセットアップされます。

## 印刷パフォーマンスの最適化

印刷パフォーマンスを最適化するには、Universal Print Serverとユニバーサルプリンタードライバーを使用します。以下のポリシー設定を構成して、印刷の最適化と圧縮を制御します。

- **ユニバーサル印刷最適化デフォルト**：セッションで作成されるユニバーサルプリンターに適用されるデフォルト設定を指定します。
  - [必要なイメージ品質] では、ユニバーサル印刷に適用されるイメージ圧縮レベルの上限を指定します。デフォルトでは [標準品質] が選択されており、ユーザーは標準品質または低品質（最大圧縮）を使ってイメージを印刷できます。
  - [ヘビーウェイト圧縮を有効にする] では、ヘビーウェイト圧縮を有効または無効にします。この機能では、画質を損なわずに [必要なイメージ品質] での圧縮レベルよりも高い帯域幅削減が提供されます。デフォルトでは、ヘビーウェイト圧縮は無効になっています。
  - [イメージおよびフォントのキャッシュ] では、印刷ストリームで使用されているイメージやフォントをキャッシュするかどうかを指定します。キャッシュを有効にすると、同一のイメージやフォントがプリンターに複数回送信されることを防ぐことができます。デフォルトでは、埋め込みイメージおよびフォントがキャッシュされます。
  - [非管理者によるこれらの設定の変更を許可する] では、非管理者ユーザーがセッション内でこれらの最適化設定を変更することを許可または禁止します。デフォルトでは、禁止されています。
- **ユニバーサル印刷イメージ圧縮制限**：ユニバーサルプリンタードライバーでのイメージ印刷で使用できる品質レベルの上限を指定します。デフォルトでは、イメージ品質の上限が [最高品質（無損失圧縮）] に設定されています。
- **ユニバーサル印刷品質制限**：セッションでの印刷出力で使用できる最大DPI値（インチあたりのドット数）を指定します。デフォルトでは、DPI値に上限はありません。

デフォルトでは、サーバーOSマシンからのすべての印刷ジョブがネットワークを介してプリントサーバーに直接転送されます。ネットワークで遅延が発生したり帯域幅に制限があったりする場合は、ICA仮想チャネルでの印刷ジョブの送信を検討します。これを行うには、Citrixポリシーの [プリントサーバーへの直接接続] 設定で [無効] を選択します。ICA仮想チャネルで送信されるデータは圧縮されるため、データがWANを横断するときに消費される帯域幅が少なくなります。

**印刷帯域幅を制限してセッションのパフォーマンスを改善する** サーバーOSマシンからユーザープリンターで印刷すると、帯域幅消費によりビデオなどほかの仮想チャネルのパフォーマンスが低下することがあります。この問題は、ユーザーが低速のネットワークを介してサーバーにアクセスする場合に顕著です。このような低下を防ぐために、ユーザープリンターでの印刷に使用される帯域幅を制限できます。転送される印刷データの量を制限すると、ビデオ、キーストローク、およびマウスデータ転送のためHDXデータストリームで使用できる帯域幅が大きくなります。

**重要**：プリンター帯域幅の制限設定は、ほかのチャネルが使用されていない場合でも常に適用されます。

Citrixポリシーで [帯域幅] カテゴリの以下の設定項目を使用して、セッションでの印刷帯域幅制限を構成します。サイトでの制限を設定するには、Studioを使ってポリシーを構成します。個々のサーバーでの制限を設定するには、各サーバーOSマシン上でローカルのWindowsグループポリシー管理コンソールを使ってポリシーを構成します。

- [プリンターリダイレクトの最大帯域幅 (Kbps)] 設定で、印刷に使用される最大帯域幅をキロビット/秒 (Kbps) で指定します。
- [プリンターリダイレクトの最大帯域幅 (%) ] 設定で、印刷に使用される最大帯域幅を、セッション全体に対する割合で指定します。

注： [プリンターリダイレクトの最大帯域幅 (%) ] 設定を使って帯域幅をパーセンテージで指定する場合は、 [セッション全体の最大帯域幅] 設定でセッション全体で使用可能な総帯域幅の最大値をキロビット/秒 (Kbps) で指定します。

最大帯域幅をKbpsおよびセッション全体に対する割合 (%) で指定した場合、より高い制限 (より低い値) の設定が適用されます。

印刷帯域幅に関する情報をリアルタイムに取得するには、Citrix Directorを使用します。

## プリンターの表示と印刷キューの管理

次の表は、プリンターを表示したり印刷キューを管理したりするためのツールの一覧です。

|                                                             | 印刷経路       | UAC<br>の有効/無効 | 場所                                                           |
|-------------------------------------------------------------|------------|---------------|--------------------------------------------------------------|
| クライアントプリンター (ユーザーデバイスに接続されたプリンター)                           | クライアント印刷経路 | オン            | Microsoft管理コンソールの [印刷の管理] スナップイン                             |
|                                                             |            | オフ            | Windows 8より古いバージョン: [コントロールパネル]<br>Windows 8: [印刷の管理] スナップイン |
| ネットワークプリンター (ネットワークプリントサーバー上のプリンター)                         | ネットワーク印刷経路 | オン            | プリントサーバーのMicrosoft管理コンソール [印刷の管理] スナップイン                     |
|                                                             |            | オフ            | プリントサーバーの [コントロールパネル]                                        |
| ネットワークプリンター (ネットワークプリントサーバー上のプリンター)                         | クライアント印刷経路 | オン            | プリントサーバーのMicrosoft管理コンソール [印刷の管理] スナップイン                     |
|                                                             |            | オフ            | Windows 8より古いバージョン: [コントロールパネル]<br>Windows 8: [印刷の管理] スナップイン |
| ローカルのネットワークサーバープリンター (サーバーOSマシンに追加されたネットワークプリントサーバー上のプリンター) | ネットワーク印刷経路 | オン            | プリントサーバーの [コントロールパネル]                                        |
|                                                             |            | オフ            | プリントサーバーの [コントロールパネル]                                        |

注: ネットワーク印刷経路で実行されたネットワークプリンターへの印刷キューは「プライベート」であり、システムで管理することはできません。

# ライセンス管理

Jan 04, 2017

## 注意

StudioとDirectorでCitrixライセンスサーバーVPXがサポートされません。CitrixライセンスサーバーVPXについて詳しくは、Citrixライセンスのドキュメントを参照してください。

管理者は、Studioを使用してライセンスを管理したり監視したりできます（ライセンスサーバーがStudioと同じドメインまたは信頼されたドメインで動作する場合）。そのほかのライセンス関連のタスクについては、Citrix eDocsの「

—製品ライセンスの有効化  
」を参照してください。

以下で説明するタスクを実行するには、すべての管理作業を実行できるライセンス管理者である必要があります。Studioでライセンス情報を表示するには、[ライセンスの表示]以上の管理者権限が必要です。組み込みのすべての管理権限を実行できる管理者と読み取り専用管理者の役割には、この権限が含まれています。

以下の表に、サポートされるエディションとライセンスモデルを示します。

| 製品         | エディション                                                                                                      | ライセンスモデル                                                                   |
|------------|-------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| XenApp     | <ul style="list-style-type: none"><li>● Platinum</li><li>● Enterprise</li><li>● Advanced</li></ul>          | 同時使用                                                                       |
| XenDesktop | <ul style="list-style-type: none"><li>● Platinum</li><li>● Enterprise</li><li>● App</li><li>● VDI</li></ul> | <ul style="list-style-type: none"><li>● ユーザー/デバイス</li><li>● 同時使用</li></ul> |

ライセンス情報を表示するには、Studioのナビゲーションペインで、[構成]、[ライセンス]の順に選択します。指定したライセンスサーバーにインストールされているすべてのライセンスの一覧と、それらのライセンスの使用状況およびサイトのライセンス設定の概要が表示されます。

ライセンス情報を管理するには、Studioのナビゲーションペインで[構成]、[ライセンス]の順に選択し、以下の操作を行います。

- Citrixからライセンスをダウンロードするには、次の手順に従います。
  1. [操作] ペインの [ライセンスの割り当て] を選択します。
  2. ライセンスアクセスコードを入力します。このコードは、Citrixからメールで送信されます。
  3. 製品を選択し、[ライセンスの割り当て] をクリックします。その製品について使用できるすべてのライセンスが割り当てられダウンロードされます。ライセンスアクセスコードを入力してすべてのライセンスを割り当ておよびダウンロードすると、そのライセンスアクセスコードは使用できなくなります。そのコードで追加のライセンス処理が必要な場合は、My Accountにログオンしてください。
- ローカルコンピューターまたはネットワークに保存されているライセンスファイルを追加するには、次の手順に従います。
  1. [操作] ペインの [ライセンスの追加] を選択します。

2. ライセンスファイルを参照して、ライセンスサーバーに追加します。
- ライセンスサーバーを変更するには、次の手順に従います。
    1. [操作] ペインの [ライセンスサーバーの変更] を選択します。
    2. ライセンスサーバーのアドレスを「:」形式で入力します。はライセンスサーバーのDNS、NetBIOS、またはIPアドレスです。ポート番号 ( ) を指定しない場合、デフォルトのポート (27000) が使用されます。
  - 使用するライセンスの種類を選択するには、次の手順に従います。
    - サイトを構成するときに、ライセンスサーバーを指定した後で、使用するライセンスの種類を選択します。サーバーにライセンスがない場合は、30日間製品を試用できるオプションが自動的に選択されます。
    - サーバーに複数のライセンスがある場合はその詳細が表示されます。いずれかのライセンスを選択します。または、サーバーにライセンスファイルを追加してそれを選択します。
  - 製品エディションおよびライセンスモデルを変更するには、次の手順に従います。
    1. [操作] ペインの [製品エディションの変更] を選択します。
    2. 適切なオプションを更新します。
  - ライセンス管理コンソールにアクセスするには、 [操作] ペインの [ライセンス管理コンソール] を選択します。ライセンス管理コンソールが自動的に開くか、パスワードによる保護が構成済みの場合は資格情報を入力するための画面が開きます。この管理コンソールの使用方法については、「  
—製品ライセンスの有効化  
」を参照してください。
  - ライセンス管理者を追加するには、次の手順に従います。
    1. 中央ペインで、 [ライセンス管理者] タブを選択します。
    2. [操作] ペインの [ライセンス管理者の追加] を選択します。
    3. 管理者として追加するユーザーを参照して、権限を選択します。
  - ライセンス管理者を編集または削除するには、管理者を選択して [操作] ペインの [ライセンス管理者の編集] または [ライセンス管理者の削除] を選択します。
    1. 中央ペインで [ライセンス管理者] タブを選択し、編集または削除する管理者を選択します。
    2. [操作] ペインの [ライセンス管理者の編集] または [ライセンス管理者の削除] を選択します。
  - ライセンス管理者グループを追加するには、次の手順に従います。
    1. 中央ペインで、 [ライセンス管理者] タブを選択します。
    2. [操作] ペインの [ライセンス管理者グループの追加] を選択します。
    3. ライセンス管理者として追加するグループを参照して、権限を選択します。Active Directoryグループを追加すると、ライセンス管理者権限がそのグループのすべてのユーザーに設定されます。
  - ライセンス管理者グループを編集または削除するには、次の手順に従います。
    1. 中央ペインで [ライセンス管理者] タブを選択し、削除または編集する管理者グループを選択します。ライセンス管理者グループを選択すると、 [操作] ペインに [ライセンス管理者グループの編集] および [ライセンス管理者グループの削除] が表示されます。
    2. [操作] ペインの [ライセンス管理者グループの編集] または [ライセンス管理者グループの削除] を選択します。

# 接続とリソース

Jan 04, 2017

管理者は、サイトを作成するときにホストリソースへの最初の接続を作成します。後でその接続を変更したり、新しい接続を作成したりできます。読み取り専用管理者は接続とリソースの詳細を表示できます。接続とリソースの管理タスクを実行するには、すべての管理権限を実行できる管理者である必要があります。

接続を作成する前に、ホストリソースが使用可能になっている必要があります。

1. Studioのナビゲーションペインで [構成]、[ホスト] の順に選択します。
2. [操作] ペインの [接続およびリソースの追加] を選択します。
3. [新しい接続を作成する] をクリックします。
4. [接続] ページで、以下の情報を指定します。
  - 接続の種類を選択し、接続名を入力します。ホストの種類やアドレスを識別するためのわかりやすい名前を指定してください。選択した接続の種類に応じて、以下の情報を指定します。

| 接続の種類                                                                                                                          | 必要な情報                                                                                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Citrix XenServer、Microsoft System Center Virtual Machine Manager、VMware vSphere、またはMicrosoft Configuration Manager Wake On LAN | 接続URL、ユーザー名、およびパスワードを入力します。 <ul style="list-style-type: none"><li>● XenServerの場合は、通信をHTTPSで保護することをお勧めします。HTTPを使用するには、XenServerと共にインストールされたデフォルトのSSL証明書を、信頼される証明機関からの証明書で置き換える必要があります。詳しくは、<a href="#">CTX128656</a>を参照してください。</li><li>● XenServerで高可用性が有効になっている場合は、新しい接続を編集して、使用する高可用性ハイパーバイザーを選択できます。</li></ul> |
| Citrix CloudPlatformまたはAmazon Web Services (AWS)                                                                               | 接続URL、APIキー、および秘密キーを入力します。 <ul style="list-style-type: none"><li>● クラウド管理者から提供されたインポートキーファイルを参照してAPIキーと秘密キーを入力できます。</li><li>● AWSコンソールから取得するルートAWSアカウント用の資格情報ファイルでは、標準的なAWSユーザーのものとは異なる形式が使用されています。このため、このファイルをStudioで使用してAPIキーと秘密キーの情報を入力することはできません。AWS環境では、StudioでAWS IAM形式の資格情報ファイルを使用してください。</li></ul>       |

- 仮想マシンの作成に使用するツールを選択します。GPUリソースを提供するハイパーバイザーの場合は、[Studioツール] を選択します。
5. [ストレージ] ページで、ストレージの種類とデバイスを選択します。Machine Creation Servicesを使用する場合は、ネットワークとストレージリソースを選択します。新しい仮想マシンを作成するときに、ここで選択したネットワークとストレージが使用されます。XenServer接続で共有ストレージを使用する場合は、IntelliCacheを有効にしてストレージデバイス上の負荷を減らすことができます。IntelliCacheの使用について詳しくは、後述の説明を参照してください。
  6. 接続にGPU機能がある場合は、グラフィック仮想化を使用するためのオプションを選択し、GPUの種類とグループを選択します。
  7. リソースの名前を入力します。

1. Studioのナビゲーションペインで [構成] 、 [ホスト] の順に選択します。
2. [操作] ペインの [接続およびリソースの追加] を選択します。
3. [既存の接続を使用する] を選択し、使用する接続を選択します。
4. 仮想マシンの作成に使用するツールを選択します。GPUリソースを提供するハイパーバイザーの場合は、 [Studioツール] を選択します。接続にGPU機能がある場合は、グラフィック仮想化を使用するためのオプションを選択し、GPUの種類とグループを選択します。
5. リソースの名前を入力します。

1. Studioのナビゲーションペインで [構成] 、 [ホスト] の順に選択します。
2. 接続を選択し、 [操作] ペインの [ストレージの追加] を選択します。
3. 追加するストレージを選択します。

1. Studioのナビゲーションペインで [構成] 、 [ホスト] の順に選択します。
2. 接続のリソースエントリを選択し、 [操作] ペインの [ストレージの編集] を選択します。
3. [標準ストレージ] ページで、仮想マシンを格納するストレージの場所のチェックボックスをオンまたはオフにします。新しいマシンの格納先として使用していたストレージの場所をオフにすると、その場所では新しいマシンが格納されなくなります。既存のマシンは引き続きその場所を使用し、その場所にデータを書き込みます。そのため、選択を解除した後もストレージの空き領域が足りなくなる場合があります。  
また、Personal vDiskストレージを使用している場合は、 [PvDストレージ] ページのチェックボックスをオンまたはオフにします。

接続の名前の変更または新しい接続の作成のために、この手順を使用しないでください。これらの操作とは異なります。

1. Studioのナビゲーションペインで [構成] 、 [ホスト] の順に選択します。
2. 接続を選択し、 [操作] ペインの [接続の編集] を選択します。
  - 接続アドレスおよび資格情報を変更するには、 [接続のプロパティ] ページで [設定の編集] をクリックし、新しい情報を入力します。  
接続のGPU設定を変更することはできません。これは、そのリソースにアクセスするマシンカタログで、GPU固有のマターイメージを使用する必要があるためです。新しい接続を作成します。
  - XenServer接続で高可用性サーバーを指定するには、 [接続のプロパティ] ページで [HAサーバーの編集] をクリックします。プールマスターに障害が生じてもXenServerとの通信が中断されないように、プール内のすべてのサーバーを選択することをお勧めします。
  - Microsoft System Center Configuration Manager (ConfMgr) のWake On LAN接続の場合は、 [詳細設定] ページで ConfMgrのウェイクアッププロキシ、マジックパケット、およびパケットの転送情報を入力します。
  - 接続での同時操作のしきい値を調整するには、以下の操作を行います。電源管理設定で同時に起動するマシンの数が多すぎたり少なすぎたりする場合にこの調整を行います。
    - [詳細設定] ページの [同時操作 (すべての種類)] と [Personal vDiskストレージインベントリの同時更新] について、この接続で同時に実行できる操作の最大数を絶対値およびこの接続を使用するすべてのマシンの数に対するパーセンテージを指定します。絶対値とパーセンテージ値の両方を指定する必要がありますが、実際にはより高い制限 (より低い値) の設定が適用されます。  
たとえば、 [同時操作 (すべての種類)] の絶対値が10、パーセンテージ値が10、この接続の総仮想マシン数が34の場合、同時に実行される操作の上限値は3 (34個のマシンのうちの10%) になります。
  - 新しい操作の1分あたりの最大数を指定します。この数は絶対値のみで指定します。



注：[詳細設定] ページの [接続オプション] ボックスへの情報の入力、Citrixサポート担当者からの指示があった場合のみ行ってください。

接続のメンテナンスモードをオンにすると、その接続（ホスト）上に格納されているマシンに新規の電源操作が適用されるのを防ぐことができます。ユーザーは、メンテナンスモードになっているマシンには接続できません。ユーザーが既に接続している場合は、そのユーザーがログオフした時点でメンテナンスモードが有効になります。

1. Studioのナビゲーションペインで [構成]、[ホスト] の順に選択します。
2. 接続を選択します。メンテナンスモードをオンにするには、[操作] ペインの [メンテナンスモードをオンにする] を選択します。メンテナンスモードをオフにするには、[メンテナンスモードをオフにする] を選択します。

個々のマシンのメンテナンスモードをオンまたはオフにすることもできます。詳しくは、後述の説明を参照してください。

注意：接続の削除は、多くのマシンおよびそのデータの損失が発生する可能性のある操作です。削除されるマシン上に重要なユーザーデータがないかどうかを確認し、重要なデータがある場合はバックアップを作成しておいてください。

接続を削除する前に、以下の点について確認してください。

- 接続上に格納されているマシンからすべてのユーザーがログオフしていること。
- 実行したまま切断されたユーザーセッションがないこと。
- プールおよび専用のマシンの場合は、メンテナンスモードになっていること。
- マシンカタログのすべてのマシンの電源がオフになっていること。

1. Studioのナビゲーションペインで [構成]、[ホスト] の順に選択します。
2. 接続を選択し、[操作] ペインの [接続の削除] を選択します。
3. この接続上にマシンが格納されている場合、マシンを削除するかどうかを確認するメッセージが表示されます。削除する場合は、それらのマシンのActive Directoryコンピューターアカウントに対する操作を指定します。  
マシンカタログで指定されている接続を削除すると、そのカタログを使用できなくなります。削除する接続がマシンカタログにより参照されている場合は、同時にそのカタログを削除することもできます。ただし、そのマシンカタログがほかの接続で使用されていないことを確認してから削除してください。

1. Studioのナビゲーションペインで [構成]、[ホスト] の順に選択します。
2. 接続を選択し、[操作] ペインの [接続名の変更] を選択します。

1. Studioのナビゲーションペインで [構成]、[ホスト] の順に選択します。
2. 接続を選択し、[操作] ペインの [マシンの表示] を選択します。

上ペインにその接続でアクセスするマシンの一覧が表示されます。マシンを選択すると、その詳細が下ペインに表示されます。実行中のセッションがある場合は、そのセッションの詳細も表示されます。

検索機能を使うと、マシンをすばやく見つけることができます。ウィンドウ上部の一覧から保存済みの検索を選択するか、または新しい検索を作成します。マシン名の一部または全体を入力して検索したり、詳細な検索式を作成したりできます。検索式を作成するには、展開ボタンをクリックして、ドロップダウンの一覧からプロパティや演算子を選択します。

1. Studioのナビゲーションペインで [構成]、[ホスト] の順に選択します。
2. 接続を選択し、[操作] ペインの [マシンの表示] を選択します。

3. [操作] ペインで、以下の管理タスクを選択します。マシンの状態や接続ホストの種類によっては、一部の操作を選択できない場合があります。

- 起動 — 電源がオフまたは一時停止状態のマシンを起動します。
- 一時停止 — マシンをシャットダウンせずに一時的に停止して、マシン一覧を更新します。
- シャットダウン — オペレーティングシステムにシャットダウンを要求します。
- 強制シャットダウン — マシンの電源を強制的に切って、マシン一覧を更新します。
- 再起動 — オペレーティングシステムに再起動を要求します。オペレーティングシステムで再起動を実行できない場合、デスクトップの状態は変更されません。
- メンテナンスモードをオンにする — マシンへの接続を一時的に無効にするために、マシンをメンテナンスモードにします。この状態のマシンにユーザーが接続することはできません。ユーザーが既に接続している場合は、そのユーザーがログオフした時点でメンテナンスモードが有効になります。

接続上のすべてのマシンのメンテナンスモードをオンまたはオフにする場合は、前述の説明を参照してください。

- デリバリーグループから削除 — マシンをデリバリーグループから削除しても、そのデリバリーグループが使用するマシンカタログからは削除されません。ユーザーが接続しているマシンは削除できません（削除するマシンにユーザーが接続しないようにするには、メンテナンスモードを一時的にオンにしてください）。
- 削除 — マシンを削除すると、ユーザーはそのマシンにアクセスできなくなります。また、そのマシンはマシンカタログから削除されます。マシンを削除する前に、必要なユーザーデータをすべてバックアップしておいてください。ユーザーが接続しているマシンは削除できません（削除するマシンにユーザーが接続しないようにするには、メンテナンスモードを一時的にオンにしてください）。

マシンのシャットダウンを伴う操作でマシンが10分以内にシャットダウンしない場合、電源が切れ、強制的にシャットダウンされます。シャットダウン中にWindowsが更新のインストールを開始すると、更新が完了する前にマシンの電源が切れる危険性があります。

1. Studioのナビゲーションペインで [構成]、[ホスト] の順に選択します。
2. リソースを選択し、[操作] ペインの [リソースの削除]、[リソース名の変更]、または [リソースのテスト] を選択します。

IntelliCacheを使用すると、共有ストレージとローカルストレージを組み合わせるようになり、VDI展開のコスト効率が向上します。これによりパフォーマンスが向上し、ネットワークトラフィックが減少します。この機能では、共有ストレージ上のマスターイメージがローカルストレージにキャッシュされ、共有ストレージでのデータ読み取りが減少します。共有デスクトップの場合、差分ディスクへの書き込みはホスト上のローカルストレージに書き込まれ、共有ストレージには書き込まれません。

- IntelliCacheを使用する場合、共有ストレージはNFSである必要があります。
- パフォーマンスを向上させるため、高パフォーマンスのローカルストレージデバイスを使用することをお勧めします。

IntelliCacheを使用するには、XenServerとStudioの両方でこの機能を有効にする必要があります。

- XenServerのインストール時に [Enable thin provisioning (Optimized storage for XenDesktop)] を選択します。IntelliCacheが有効なサーバーと無効なサーバーを同一プールで混在させることはサポートされません。詳しくは、XenServerのドキュメントを参照してください。
- XenAppおよびXenDesktopでは、IntelliCacheはデフォルトで無効になっています。この機能はXenServer接続の作成時のみ有効にでき、これを後で無効にすることはできません。StudioでXenServer接続を追加するときに、以下の手順に従います。
  - ストレージの種類として、[共有] を選択します。
  - [IntelliCacheを使用して共有ストレージデバイス上の負荷を軽減させる] チェックボックスをオンにします。

ポリシー設定を使用すると、以下の3つの接続タイマーを構成できます。

- 最長接続タイマー：このタイマーでは、ユーザーデバイスとデスクトップ間の連続セッションを自動的にログオフするまでの時間を制御します。これを構成するには、ポリシーの [セッション接続タイマー] 設定および [セッション接続タイマー間隔] 設定を使用します。
- 接続アイドルタイマー：このタイマーでは、ユーザーからの入力がないユーザーデバイスとデスクトップ間の連続セッションを自動的にログオフするまでの時間を制御します。これを構成するには、ポリシーの [セッションアイドルタイマー] 設定および [セッションアイドルタイマーの間隔] 設定を使用します。
- 切断タイマー：このタイマーでは、切断状態でロックされたデスクトップセッションを自動的にログオフするまでの時間を制御します。これを構成するには、ポリシーの [切断セッションタイマー] 設定および [切断セッションタイマーの間隔] 設定を使用します。

これらの設定項目を変更する場合は、環境全体で設定が一貫していることを確認してください。

# 接続リース

Jan 04, 2017

サイトデータベースを常に使用可能状態にするために、Microsoft社の高可用性ベストプラクティスに従って耐障害性の高いSQL Server展開から開始することをお勧めします。ただし、ネットワークの問題および中断によってDelivery Controllerがデータベースにアクセスできなくなり、その結果、ユーザーがアプリケーションやデスクトップに接続できなくなる場合があります。

SQL Server高可用性ベストプラクティスを補完する接続リース機能を使用すると、サイトデータベースが使用不可の場合でも、ユーザーが直近で使用したアプリケーションやデスクトップに接続および再接続できるようになります。

通常、多くの公開リソースを使用できるユーザーでも、定期的に使用するのはその一部のみです。接続リース機能を有効にすると、通常時（データベースが使用可能なとき）に、最近使用されたアプリケーションやデスクトップのユーザー接続が各Controllerでキャッシュされます。

各Controller上でリースが生成され、それがサイトデータベースにアップロードされてほかのControllerと定期的に同期されます。各Controllerのキャッシュには、リースのほかにアプリケーション、デスクトップ、アイコン、ワーカーなどの情報が格納されます。リースとその関連情報は、各Controllerのローカルディスクに保存されます。データベースを使用できなくなると、Controllerはリース接続モードになります。この場合、ユーザーがStoreFrontから最近使用したアプリケーションやデスクトップへの接続を試みると、キャッシュされている操作が「再生」されます。

接続は、2週間の「リース期間」をキャッシュされます。そのため、ユーザーが直前の2週間に起動したデスクトップとアプリケーションは、データベースが使用不可になった場合でも引き続きStoreFront経由でアクセスできます。ただし、直前の2週間に起動しなかったデスクトップとアプリケーションは、データベースが使用不可になった場合にはアクセスできなくなります。たとえば、あるアプリケーションを最後に起動したのが3週間前である場合、そのアプリケーションのリース期限が切れているため、現時点でデータベースが使用不可になると起動できなくなります。長時間実行されているアプリケーションまたはデスクトップのアクティブセッションまたは切断セッションのリースは、期限切れにならないように延長されます。

デフォルトでは、接続リースはサイト全体に影響します。ただし、特定のユーザーのすべてのリースを取り消すことができます。これにより、Controllerがリース接続モードのときに、そのユーザーはすべてのアプリケーションやデスクトップにアクセスできなくなります。そのほかのいくつかのレジストリ設定はController単位で適用されます。

接続リースを使用すると接続の回復性やユーザーの生産性を向上させることができますが、そのほかの機能のパフォーマンス、可用性、および操作に関連する注意事項があります。

接続リースは、サーバーでホストされるアプリケーションおよびデスクトップと静的な（割り当て済み）デスクトップでサポートされます。プール型のVDIデスクトップや、データベースが利用できなくなった時点でデスクトップが割り当てられていないユーザーはサポートされません。

リース接続モードのControllerでは、以下の制限事項があります。

- 管理者はStudio、Director、またはPowerShellコンソールを使用できません。
- ワークスペースコントロールは使用できません。ユーザーがReceiverにログオンしたときにセッションには自動的に再接続されません。ユーザーはアプリケーションを再度起動する必要があります。
- すべてのController間でリース情報が同期されるまでは、データベースが利用できなくなったときにユーザーがそのリソースの起動に失敗する場合があります。
- サーバーでホストされるアプリケーションとデスクトップのユーザーは、構成されている最大セッション数よりも多くのセッションを使用できる場合があります。次に例を示します。

- Controllerがリース接続モードになっていないときにユーザーがデバイス（NetScaler Gateway経由で外部から接続しているデバイス）からセッションを起動し、Controllerがリース接続モードになった後でLAN上の別のデバイスから接続した場合、セッションがローミングされずに新しいセッションが作成されることがあります。
- データベースが使用できなくなる直前にアプリケーションが起動した場合、セッションの再接続に失敗することがあります。この場合、新しいセッションとアプリケーションインスタンスが起動します。
- 静的な（割り当て済み）デスクトップでは、電源管理が行われません。Controllerがリース接続モードになるときに電源がオフになったVDAは、管理者が手動で電源をオンにしない限り、データベース接続が回復するまで使用できません。
- セッションの事前起動とセッション残留が有効になっている場合、新しい事前起動セッションは開始されません。データベースが利用可能になるまでは、構成されたしきい値に従って事前起動セッションや残留セッションが自動終了することはありません。
- サイト内の負荷管理が影響を受ける可能性があります。サーバーベースの接続は直近に使用されたVDAに割り当てられません。負荷評価基準（特にセッション数規則）を超過する可能性があります。
- SQL Server Management Studioと使ってデータベースをオフラインにすると、Controllerがリース接続モードになりません。以下のTransact-SQL文を使用してください。
  - ALTER DATABASE <database-name> SET OFFLINE WITH ROLLBACK IMMEDIATE
  - ALTER DATABASE <database-name> SET OFFLINE WITH ROLLBACK AFTER <seconds>
 これらのSQL文によりすべての保留中のトランザクションがキャンセルされ、Controllerとデータベースとの接続が失われます。これにより、Controllerがリース接続モードになります。

接続リースが有効になっている場合、ユーザーが接続または再接続できない短い間隔が2回あります。1回はデータベースが不具合でなくなってからControllerがリース接続モードになるまでの間、もう1回はControllerがリース接続モードを終了してからデータベースが完全に回復し、VDAの再登録が終わるまでの間です。

このほかの考慮事項については、「[XenDesktop 7.6の接続リースの設計上の考慮事項](#)」を参照してください。

接続リース機能を使用するには、環境を以下のように構成する必要があります。

- VDAはVersion 7.6以上である必要があります。また、これらのマシンを使用するマシンカタログとデリバリーグループもVersion 7.6以上である必要があります。
- サイトデータベースのサイズ要件が大きくなります。
- 各Controllerには、キャッシュされるリースファイル用の追加のディスクスペースが必要です。

接続リース機能は、デフォルトで有効になっています。

接続リース機能の有効/無効は、PowerShell SDK、またはWindowsのレジストリで切り替えることができます。PowerShell SDKを使用する場合は、現在のリースを削除することもできます。接続リース機能を制御するPowerShellコマンドレットは以下のとおりです。詳しくは、コマンドレットのヘルプを参照してください。

- Set-BrokerSite -ConnectionLeasingEnabled \$true|\$false — 接続リース機能のオン/オフを切り替えます。デフォルト値：\$true
- Get-BrokerServiceAddedCapability - ローカルのControllerに「ConnectionLeasing」を出力します。
- Get-BrokerLease - 現在のリースを取得します。フィルターで指定したリースのみを取得することもできます。
- Remove-BrokerLease - 指定したリースを削除用にマーク付けします。フィルターで指定したリースのみをマーク付けすることもできます。
- Update-BrokerLocalLeaseCache — ローカルのController上の接続リースのキャッシュを更新します。データは次の同期時に再同期されます。

# 仮想IPおよび仮想ループバック

Jan 04, 2017

注：これらの機能は、Windows Server 2008 R2マシンおよびWindows Server 2012 R2マシンでのみ有効です。WindowsデスクトップOSマシンでは使用できません。

Microsoft社の仮想IPアドレス機能により、セッションごとに動的に割り当てられるIPアドレスを公開アプリケーションで使用できます。Citrixの仮想ループバック機能を使用すると、ローカルホスト（デフォルトで127.0.0.1）と通信するアプリケーションで、ローカルホストの範囲内（127.\*）で固有の仮想ループバックアドレスが使用されるように構成できます。

CRM（Customer Relationship Management）やCTI（Computer Telephony Integration）などの特定のアプリケーションでは、アドレス割り当て、ライセンス付与、識別、またはそのほかの目的でIPアドレスが使用されるため、セッションに固有のIPアドレスまたはループバックアドレスが必要です。また、一部のアプリケーションでは静的なポートにバインドされるため、マルチユーザー環境でそのアプリケーションの追加インスタンスを起動しようとする、そのポートが使用済みなので、動的に失敗します。これらのアプリケーションがXenApp環境で正しく動作するためには、クライアントデバイスごとに異なるIPアドレスが使用される必要があります。

仮想IPと仮想ループバックは、それぞれ独立した機能です。これらの機能のいずれかまたは両方を使用できます。

使用する機能に応じて、管理者は以下の操作を行います。

- Microsoft社の仮想IP機能を使用するには、Windowsサーバー上で仮想IPを有効にして構成します。
- Citrixの仮想ループバック機能を使用するには、Citrixポリシーで2つの設定項目を構成します。

Windowsサーバー上で仮想IP機能を有効にすると、セッション内で動作する各アプリケーションで固有のアドレスが使用されるように構成できます。ユーザーは、これらのアプリケーションを、ほかの公開アプリケーションと同じように使用することができます。以下のいずれかの動作をするプロセスでは、仮想IPアドレスを設定します。

- ハードコードされた（固定された）TCPポート番号を使用する。
- Windowsソケットを使用し、固有のIPアドレスまたは固定されたTCPポート番号を使用する。

アプリケーションで仮想IPアドレスが必要かどうかを判断するには、次の手順に従います。

1. Microsoft社のWebサイトから、TCPViewツールを入手します。このツールを使用すると、特定のIPアドレスおよびポートを使用しているすべてのアプリケーションを一覧表示できます。
2. TCPViewの [Options] メニューで、 [Resolve Addresses] を無効にします。これにより、一覧にホスト名ではなくアドレスが表示されるようになります。
3. 対象となるアプリケーションを起動して、使用されているIPアドレスとポート、およびそれらのポートを開いているプロセスの名前をTCPViewで確認します。
4. サーバーのIPアドレス0.0.0.0または127.0.0.1を使用するプロセスを構成します。
5. そのアプリケーションの追加インスタンスを起動して、別のポート上で同じIPアドレスが使用されないことを確認します。

## Microsoftリモートデスクトップ (RD) のIP仮想化機能のしくみ

- 仮想IPアドレスを使用するには、Windowsサーバー上でこの機能を有効にする必要があります。  
たとえば、Windows Server 2008 R2環境でサーバーマネージャーを使用し、 [リモートデスクトップサービス] の [RDセッションホストの構成] を展開してRD IP仮想化機能を有効にします。次に、IPアドレスをDHCP（Dynamic Host Configuration Protocol：動的ホスト構成プロトコル）サーバーによりセッションごとまたはプログラムごとに動的に割り当てるように設定を行います。手順については、Microsoft社のドキュメントを参照してください。
- この機能を有効にすると、セッション起動時にサーバーは、DHCPサーバーから動的に割り当てられたIPアドレスを要求します。

- RD IP仮想化機能によって、セッションごとまたはプログラムごとに、リモートデスクトップ接続にIPアドレスが割り当てられます。複数のプログラムにIPアドレスを割り当てる場合、これらのプログラム間でセッションごとのIPアドレスが共有されます。
- アドレスが割り当てられたセッションでは、bind、closesocket、connect、WSAConnect、WSAAccept、getpeername、getsockname、sendto、WSASendTo、WSASocketW、gethostbyaddr、getnameinfo、getaddrinfoの各コールに対して、システムのプライマリIPアドレスではなく仮想アドレスが使用されます。

リモートデスクトップセッションのホスト環境でMicrosoftのIP仮想化機能を使用すると、アプリケーションとWinsockコールとの間に「フィルター」コンポーネントを挿入することで、アプリケーションと特定のIPアドレスがバインドされます。IPアドレスがバインドされると、アプリケーションはそのアドレスだけで要求を待ち受けるようになります。アプリケーションのTCPリスナーまたはUDPリスナーは自動的に仮想IPアドレス（または仮想ループバックアドレス）にバインドされ、アプリケーションからの接続はその仮想アドレスから開かれます。

Windowsポリシーにより制御されるGetAddrInfo()など、アドレスを返すファンクションでローカルホストIPアドレスが要求されると、返されたIPアドレスがそのセッションの仮想IPアドレスに変換されます。このようなファンクションでローカルサーバーのIPアドレスを取得しようとするアプリケーションには、セッション固有の仮想IPアドレスだけが渡されます。このようにしてアプリケーションに渡されたIPアドレスは、後続のソケットコール（bindやconnectなど）で使用されます。

アプリケーションでは、アドレス0.0.0.0で、リスナー用のポートのバインドが必要になる場合があります。このようなアプリケーションで静的なポート番号が使用されると、競合が発生するため、複数のインスタンスを起動できなくなります。仮想IPアドレス機能では、0.0.0.0へのファンクションコールが特定の仮想IPアドレスに変換されます。これにより、セッションごとに異なるアドレス上のポートが使用されるため、同じポート番号を使用する複数のアプリケーションを実行できるようになります。このファンクションコールは、仮想IPアドレス機能が有効なICAセッションでのみ変換されます。たとえば、すべてのインターフェイス（0.0.0.0）と特定のポート（9000など）にバインドするアプリケーションの2つのインスタンスが、それぞれ異なるセッションで実行される場合、VIPAddress1:9000とVIPAddress2:9000にバインドされるため、競合が起きません。

Citrixポリシーで仮想IPループバック機能を有効にすると、各セッションで通信に独自のループバックアドレスが使用されるようになります。アプリケーションがWinsockコールでローカルホストのアドレス（デフォルトで127.0.0.1）を使用する場合、仮想ループバック機能により、127.0.0.1が127.X.X.X（X.X.XはセッションIDに1を加えた値）に置き換えられます。たとえば、セッションIDが7の場合は127.0.0.8になります。セッションIDが4オクテットを超える場合（つまり255を超える場合）は、127.0.1.0のように次のオクテットに繰り上げられます。また、最大値は127.255.255.255です。

以下のいずれかの動作をするプロセスでは、仮想ループバックを設定します。

- Windowsソケットのループバック（localhost）アドレス127.0.0.1を使用する。
- ハードコードされた（固定された）TCPポート番号を使用する。

プロセス間通信でループバックアドレスを使用するアプリケーションでは、[仮想ループバックアドレスに関するポリシー設定](#)を使用します。追加の構成は必要ありません。仮想ループバックは仮想IPに依存しないため、Windowsサーバーの構成は不要です。

- 仮想IPループバックサポート：このポリシー設定を有効にすると、各セッション固有の仮想ループバックアドレスが使用されるようになります。このチェックボックスは、デフォルトでオフになっています。この機能は、[仮想IPループバックプログラム一覧] ポリシー設定で指定したアプリケーションにのみ適用されます。
- 仮想IPループバックプログラム一覧：このポリシー設定では、仮想IPループバック機能を使用するアプリケーションを指定します。この設定は、[仮想IPループバックサポート] ポリシー設定が有効になっている場合のみ適用されます。

## 関連機能

次のレジストリ設定により、仮想ループバックが仮想IPよりも優先されるようになります（優先ループバック機能）。ただ

し、以下の点に注意してください。

- 優先ループバックはWindows 2008 R2でのみサポートされます。
- 仮想IPアドレスと仮想ループバックの両方の機能を有効にする場合にのみ、優先ループバック機能を使用してください。そうしないと、意図しない結果が生じる可能性があります。
- レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

アプリケーションのホストサーバー上で、regeditを実行します。

- HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\VIP (32ビットマシンでは HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\VIP)
- 値の名前：PreferLoopback、種類：REG\_DWORD、値のデータ：1
- 値の名前：PreferLoopbackProcesses、種類：REG\_MULTI\_SZ、値のデータ：<list of processes>。ここで、はプロセスの一覧です。



# セカンダリデータベースの場所

Jan 04, 2017

デフォルトでは、構成ログおよび監視データベース（セカンダリデータベース）はサイト構成データベースと同じサーバーに置かれます。初期状態では、これらのデータベースに同じ名前が設定されます。サイトを作成した後で、セカンダリデータベースの場所を変更することをお勧めします。構成ログデータベースと監視データベースは、同じサーバー上に配置したり別のサーバー上に配置したりできます。各データベースを異なる方法でバックアップしなければならない場合があります。

構成ログデータベースと監視データベースの場所を変更する場合は、次の点に注意してください。

- 変更前のデータベース内のデータは変更後のデータベースにインポートされません。
- 構成ログデータベースの場所を変更する場合、変更前のデータベースの内容は集約されなくなります。
- 変更後のデータベースの最初にデータベースの変更を示すログが記録されますが、変更前のデータベースの場所は記録されません。

構成ログデータベースまたは監視データベースの場所を変更する前に、変更後のサーバー上に適切なバージョンのMicrosoft SQL Serverをインストールします。また、必要に応じて、ミラー、クラスター、またはそのほかの冗長インフラストラクチャをセットアップします。

データベースが切断されているときの構成変更が禁止された環境（必須ログ機能）では、構成ログの場所を変更することはできません。

注：サイト構成データベースの場所は、この方法で変更することはできません。

1. Studioのナビゲーションペインで [構成] を選択します。3つのデータベースの名前とアドレスが一覧で表示されます。ミラーサーバーが構成されている場合は、そのアドレスも表示されます。
2. 場所を変更するデータベースを選択して、[操作] ペインの [データベースの変更] を選択します。
3. 次の表の形式で、新しいSQL Serverデータベースの場所とデータベース名を指定します。

| データベースの種類     | 入力内容                        | データベースの構成                                                 |
|---------------|-----------------------------|-----------------------------------------------------------|
| スタンドアロンまたはミラー | servername                  | デフォルトのインスタンスが使用され、SQL Serverでデフォルトのポートが使用されている場合。         |
|               | Servername\INSTANCENAME     | 名前付きのインスタンスが使用され、SQL Serverでデフォルトのポートが使用されている場合。          |
|               | servername,port-number      | デフォルトのインスタンスが使用され、SQL Serverでカスタムポートが使用されている場合。（コンマが必要です） |
| その他           | cluster-name                | データベースがクラスター化されている場合。                                     |
|               | availability-group-listener | Always-Onデータベースが構成されている場合。                                |

4. 新しいデータベースをStudioで作成する場合は、[OK] をクリックします。確認のメッセージが表示され、[OK] をクリックするとStudioによりデータベースが自動的に作成されます。現在のStudioユーザーの資格情報を使ってデータベースへのアクセスが試行されます。アクセスに失敗すると、データベースにアクセスするための資格情報を入力する画面がきます。アクセスに成功すると、データベーススキーマがデータベースにアップロードされます（資格情報はデータベース作成時のみ保持されます）。

5. データベースを手動で作成する場合は、[データベーススクリプトの生成] をクリックします。作成されるスクリプトには、データベースおよびミラーデータベース（構成する場合）を手動で作成するためのコマンドが記述されます。スキーマをアップロードする前に、データベースが空であること、および1人以上のユーザーがそのデータベースにアクセスでき、変更できることを確認してください。

# Delivery Controller環境

Jan 04, 2017

Delivery Controllerは、ユーザーアクセスの管理や接続の仲介と最適化を行うためのサーバー側のコンポーネントです。また、Controllerは、デスクトップおよびサーバーイメージを作成するMachine Creation Serviceも提供します。

サイトには、1つ以上のDelivery Controllerが必要です。まずControllerをサーバーにインストールしてサイトを作成し、次に1つ以上のControllerをそのサイトに追加します。サイトに複数のControllerがあると、以下の2つの利点をもたらされます。

- 冗長性 — 実務サイトでのベストプラクティスとして、物理サーバー上で動作するControllerを2台以上配置します。一方のControllerに障害が発生しても、他方のControllerで接続を管理し、サイトを制御できます。
- スケーラビリティ — サイトのアクティビティが増えるにつれ、Controller上のCPU使用量およびSQL Serverデータベースのアクティビティも増えます。Controllerを追加すると、より多くのユーザーやより多くのアプリケーションやデスクトップ要求を処理できるようになり、制御処理全体を向上させることができます。

VDAを使用するには、そのサイトのControllerに登録（接続を確立）する必要があります。VDAはListOfDDCsと呼ばれるControllerの一覧をチェックしてControllerを見つけます。ListOfDDCsは、そのサイトのControllerのDNSエントリまたはIPアドレスで構成されています。負荷分散のため、VDAは一覧のすべてのControllerで接続を自動的に分散させます。

ListOfDDCsに加えて、ListOfSIDsと呼ばれるControllerのマシンセキュリティID (SID) 一覧により、VDAがアクセスできるControllerが指定されます。ListOfSIDsは、Active Directoryでの負荷を軽減したり、改ざんされたDNSサーバーからのセキュリティ上の脅威を防いだりするために使用できます。

サイト内でControllerの追加や削除に伴って、すべてのVDA上のListOfDDCsおよびListOfSIDsが確実に更新されるようにすることが重要です。これらの一覧を正しく更新しない場合、一覧にないControllerで仲介されたセッションがVDAで拒否される可能性があります。また、無効なエントリにより、仮想デスクトップシステムソフトウェアの起動に遅延が生じることがあります。これらの一覧は、以下の方法で更新できます。

- 自動更新 — Controllerが追加または削除されたときに、ListOfDDCsおよびListOfSIDsを自動的に更新します。デフォルトでは、自動更新が有効になっています。
- 自己管理 — Controllerを識別するためのポリシーやレジストリ設定を手動で更新します。

ListOfDDCsおよびListOfSIDsに含まれている情報は、展開内の複数の場所で参照されます。VDAは以下の場所を順にチェックして、最初に見つかった一覧を使用します。

1. 自動更新機能により維持される永続的なストレージ。自動更新が有効な場合、VDAのインストール直後の初回登録以降では、ここにController情報が格納されます（このストレージには、マシンポリシーの情報も格納されます。これにより、再起動後もポリシー設定が保持されます）。インストール直後の初回登録時、または自動更新が無効な場合、VDAは以下の場所をチェックします。
2. ポリシー設定 (Controller、Controller SID) 。
3. レジストリのVirtual Desktop Agentキーの下でのController情報。これらの値は、VDAのインストール時に指定するController情報に基づいて、VDAのインストーラーにより設定されます。
4. 組織単位ベースのController検出。これは、後方互換性を維持するための従来の方法です。
5. Machine Creation Servicesにより作成されるPersonality.iniファイル。

ListOfDDCsに複数のControllerが指定されている場合、VDAはランダムな順序で接続を試行します。ListOfDDCsには、複数のControllerのエントリを角かっことで囲んだControllerグループを含めることもできます。VDAは、これらのグループ内の各Controllerへの接続を試行し、その後でほかのエントリを試行します。

Version 7.0より前のXenDesktopをアップグレードした場合は、従来のCNAME機能が自動更新機能に置き換えられます。必要な場合は手動でCNAME機能を再度有効にすることができますが、DNSエイリアスの整合性に問題が生じるため、自動更新機能とCNAME機能の両方を使用することはできません。CNAME機能の再有効化について詳しくは、[CTX139555](#)を参照してください。

自動更新機能は、ポリシー設定によりデフォルトで有効になっています。

以下の様なサイトでは自動更新を使用できないため、自己管理を行う必要があります。

- Controllerグループを使用する展開。
- セキュリティ上の理由によりListOfSIDsを使用する展開（Active Directoryの負荷軽減のためにListOfSIDsを使用する場合は自動更新を使用できます）。
- ライトバックディスクなしでProvisioning Servicesを使用する展開。
- [Controller] または [コントローラーSID] ポリシー設定を使用する展開。

自動更新のポリシー設定は、[Virtual Delivery Agent設定] カテゴリにあります。

- 自動更新を有効にするには、ポリシーの[Controllerの自動更新を有効にする]設定を有効にします。この設定項目は、デフォルトで有効になっています。
- 自動更新を無効にするには、[Controllerの自動更新を有効にする]設定を無効にします。

自動更新が有効な環境でVDAをインストールする場合、VDAのインストール時に指定したいずれかのControllerにVDAが登録されます。VDAのインストール時に指定したController情報は、インストーラーによりListOfDDCsレジストリ値に書き込まれます。

VDAをControllerに登録すると、登録先のControllerにより環境内のControllerの完全修飾ドメイン名 (FQDN) およびセキュリティID (SID) の一覧がVDAに送信されます。この一覧の内容は、VDAにより自動更新用の永続的なストレージに書き込まれます。また、各Controllerは90分ごとにサイトのデータベースにアクセスして、Controllerの追加や削除、およびポリシーの変更内容について確認し、登録したVDAに更新情報を送信します。VDAは、受信した最新の一覧に基づいてすべてのControllerからの接続を受け入れます。

VDAが受信した一覧に登録先のControllerが含まれていない場合（つまりそのControllerがサイトから削除された場合）、一覧のいずれかのControllerにVDAが再登録されます。VDAの登録または再登録が完了すると、新しい一覧を受信します。

次に例を示します。

1. 環境内に3つのControllerであるA、B、Cがあります。VDAをインストールしたときに、そのVDAが（インストール時に指した）Controller Bに登録されます。
2. サイトに2つのController（DおよびE）を追加します。90分以内に、更新された一覧がVDAに送信されます。これにより、VDAはController A、B、C、D、Eからの接続を受け入れるようになります（VDAを再起動するまでは、すべてのController間で負荷分散は行われません）。
3. サイトからController Bを削除します。前回のチェック以降にサイトのControllerに変更があったため、90分以内にVDAは更新済みの一覧を受信します。手順1でインストールしたVDAはController Bに登録されていましたが、このControllerは一覧から削除されているため現在のController（A、C、D、E）のいずれかに再登録されます。

自動更新機能を使用しない場合は、サイトのDelivery Controllerを追加、移動、または削除した後で各Virtual Delivery Agent (VDA) またはそのイメージのレジストリ値またはCitrixポリシー設定を管理者が更新する必要があります。レジストリの変更は、グループポリシーオブジェクトを使って更新することもできます。

Citrixポリシー設定を使って自己管理するには

1. ポリシーの [Controller] 設定で指定した完全修飾ドメイン名値を更新します。このポリシー設定は、 [Virtual Delivery Agent設定] カテゴリにあります。
2. ListOfSIDsを併用している環境では、 [コントローラーSID] 設定で指定したSID値を更新します。

レジストリを使って自己管理するには

注意：レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

1. サイトのすべてのControllerの完全修飾ドメイン名の一覧が設定されているListOfDDCsレジストリキーを更新します（このキーはActive Directoryサイト組織単位に相当します）。スペースで区切って複数の値を入力できます。Controllerグループは角かっこで囲みます。

HKEY\_LOCAL\_MACHINE\Software\Citrix\VirtualDesktopAgent\ListOfDDCs (REG\_SZ)

HKEY\_LOCAL\_MACHINE\Software\Citrix\VirtualDesktopAgentにListOfDDCsとFarmGUIDキーの両方がある場合は、ListOfDDCsがControllerの検出に使用されます。FarmGUIDは、VDAのインストール時にサイト組織単位を指定した場合に作成されます。

2. 必要に応じて、ListOfSIDsレジストリキーを更新します。

HKEY\_LOCAL\_MACHINE\Software\Citrix\VirtualDesktopAgent\ListOfSIDs (REG\_SZ)

# Controllerの追加、削除、または移動、またはVDAの移動

Jan 04, 2017

Delivery Controllerを追加、削除、または移動するには、以下のロールまたは権限が必要です。

| 操作                       | 目的                                                   | サーバーロール            | データベースロール |
|--------------------------|------------------------------------------------------|--------------------|-----------|
| データベースの作成                | 空のデータベースを作成します                                       | dbcreator          |           |
| スキーマの作成                  | サービス固有のすべてのスキーマを作成して、サイトに最初のControllerを追加します         | securityadmin<br>* | db_owner  |
| Controllerの追加            | サイトにController (2つ目以降) を追加します                        | securityadmin<br>* | db_owner  |
| Controller (ミラーサーバー) の追加 | ミラー化されたデータベースのミラーロールのデータベースサーバーにControllerログインを追加します | securityadmin<br>* |           |
| Controllerの削除            | サイトからControllerを削除します                                |                    | db_owner  |
| スキーマの更新                  | スキーマの更新およびHotfixを適用します                               |                    | db_owner  |

\* securityadminは、技術的にはより限定的なサーバーロールですが、実際にはsysadminサーバーロールと同等のものとして扱われます。

Studioを使ってこれらの操作を実行する場合、ユーザーアカウントが明示的にsysadminサーバーロールのメンバーである必要があります。

展開環境でデータベースのミラーリングを使用している場合は、以下の点について注意してください。

- Controllerを追加、削除、または移動する前に、プライマリデータベースとミラーデータベースの両方が実行中であることを確認してください。また、SQL Server Management Studioでスクリプトを使用している場合は、SQLCMDモードを有効にしてください。
- Controllerを追加、削除、または移動した後でミラーリング設定を検証するには、PowerShellコマンド `Get-ConfigDbConnection` を実行して、ミラーに対する接続文字列でフェールオーバーパートナーが設定されていることを確認します。

Controllerの追加、削除、または移動後の作業

- 自動更新が有効な場合は、Virtual Delivery Agent (VDA) が90分以内に最新のController一覧を受信します。
- 自動更新が無効な場合は、すべてのVDAについてControllerポリシー設定またはListOfDDCsレジストリキーが更新されていることを確認してください。Controllerをほかのサイトに移動した後は、両方のサイト上でポリシー設定またはレジストリキーを更新する必要があります。

以前のバージョンがインストールされたサーバーをこのバージョンのサイトに追加することはできません。

1. 追加するサーバー上でインストーラーを実行し、Delivery Controllerおよびそのほかのコアコンポーネントを選択します。
2. Studioで [既存の展開に参加] をクリックし、既存のサイトのアドレスを入力します。

Controllerを削除すると、Citrixソフトウェアやそのほかのコンポーネントはアンインストールされませんが、データベースからそのControllerが削除されます。このため、このControllerでは接続の仲介やそのほかのタスクを実行できなくなります。削除したControllerを、後で元のサイトや別のサイトに追加することができます。サイトには最低1つのControllerが必要なため、Studioの一覧に表示される最後のControllerを削除することはできません。

注：Controllerが動作しており、1時間以内にそのControllerがStudioにロードされることを確認してください。削除するControllerがStudioにロードされたら、メッセージに従ってControllerをシャットダウンしてください。

サイトからController削除しても、データベースサーバーへのControllerログオンは削除されません。これは、同じマシン上のほかの製品のサービスで使用されるログオンが削除されるのを防ぐためです。ログオンが必要ない場合には、手動で削除する必要があります。ログオンの削除には、securityadminサーバーロール権限が必要です。

**重要：**サイトからControllerを削除するまでは、Active DirectoryでそのControllerを削除しないでください。

1. Studioのナビゲーションペインで、[構成]、[Controller] の順に選択し、削除するControllerを選択します。
2. [操作] ペインの [Controllerの削除] を選択します。適切なデータベースロールや権限がない場合は、Controllerを削除するためのスクリプトを生成できます。そのスクリプトの実行をデータベース管理者に依頼してください。
3. データベースサーバーからControllerのマシンアカウントを削除しなければならない場合があります。これを行う前に、ほかのサービスがそのアカウントを使用していないことを確認してください。

Studioを使ってControllerを削除した後、実行中のタスクを適切に完了させるためにそのControllerへのトラフィックがしばらく残ることがあります。Controllerを即座に削除するには、Controllerがインストールされているサーバーをシャットダウンするか、Active Directoryからそのサーバーを削除することをお勧めします。その後で、サイト内のほかのControllerを再起動します。これにより、削除されたControllerとの通信が行われなくなります。

このソフトウェアの以前のバージョンで作成されたサイトには、Controllerを移動できません。

1. StudioのナビゲーションペインでControllerの移動元サイトを開き、[構成]、[Controller] の順に選択し、移動するControllerを選択します。
2. [操作] ペインの [Controllerの削除] を選択します。適切なデータベースロールや権限がない場合は、Controllerを削除するためのスクリプトを生成できます。そのスクリプトの実行をデータベース管理者に依頼してください。サイトには最低1つのControllerが必要なため、Studioの一覧に表示される最後のControllerを削除することはできません。
3. 移動するControllerでStudioを開き、確認メッセージに応じてサービスをリセットします。さらに、[既存のサイトへ参加] を選択して、移動先サイトのアドレスを入力します。

VDAがProvisioning Servicesを使ってプロビジョニングされた場合、または既存のイメージの場合、アップグレード時、またはテストサイトで作成されたVDAイメージを実務環境サイトに移動させる場合に、VDAをほかのサイトに移動（サイト1からサイト2へ）できます。MCSはListOfDDCsの変更をサポートせずVDAはControllerへの登録をチェックするため、Machine Creation Services (MCS) を使ってプロビジョニングされたVDAをあるサイトから別のサイトには移動できません。MCSを使ってプロビジョニングされるVDAは、作成されたサイトに割り当てられたListOfDDCsをチェックします。

VDAをほかのサイトに移動するにはインストーラーを使用するか、Citrixポリシーを使用します。

- インストーラー：インストーラーを実行して、サイト2のControllerのFQDN（DNSエントリ）を指定してこのControllerを

追加します。

重要：Controllerのポリシー設定を使用しない場合にのみ、インストーラーでControllerを指定してください。

- グループポリシーエディター：次の例では、複数のVDAをほかのサイトに移動します。
  1. サイト1でポリシーを作成して以下のように設定し、そのポリシーをVDA移行を行うデリバリーグループに割り当てます。
    - Controller — サイト2のいくつかのControllerのFQDN（DNSエントリ）を指定します。
    - Controllerの自動更新を有効にする — 無効に設定します。
  2. デリバリーグループの各VDAは、新しいポリシーの適用後90分以内にアラートを受信します。VDAは、受信したControllerの一覧を無視して（自動更新が無効なため）、ポリシーで指定されているサイト2のいずれかのControllerを選択します。
  3. VDAがサイト2のControllerへの登録に成功すると、サイト2のListOfDDCsおよびポリシー情報を受け取って、これにより自動更新が有効になります。サイト1での登録先のControllerがサイト2のControllerによって送信された一覧にはないため、サイト2の一覧のControllerのいずれかにVDAが再登録されます。これにより、VDAはサイト2からの情報に基づいて自動的に更新されます。



# Active Directory OUベースのController検出

Jan 04, 2017

このDelivery Controller検出方法は主に後方互換性のためにサポートされており、VDA for Windows Desktop OSでのみ有効です（VDA for Windows Server OSでは使用できません）。この検出方法では、サイト内のすべてのコンピューターが同じActive Directoryドメインに属しており、Controllerとデスクトップで使用されるドメイン間に相互信頼関係が設定されている必要があります。この検出方法を使用する場合、各デスクトップのレジストリで組織単位（OU）のGUIDを構成する必要があります。

OUベースのController検出を実行するには、Controller上でPowerShellスクリプトのSet-ADControllerDiscovery.ps1を実行します（このスクリプトは各Controllerの\$Env:ProgramFiles\Citrix\Broker\Service\Setup Scriptsフォルダーにあります）。スクリプトを実行するには、すべての管理タスクを実行できる権限に加えて、親OUでのCreateChild権限が必要です。

Active Directoryを使用してControllerが検出されるように構成する場合は、サイトを作成するときに、対応する組織単位（OU）をActive Directory内に作成する必要があります。このOUは、環境内のコンピューターが属しているフォレスト内のあるドメインにも作成できます。ベストプラクティスとしては、サイト内のControllerもこのOUに含まれている方が望ましいですが、これは必須ではありません。適切な権限を持つドメイン管理者はOUを空のコンテナとして作成して、そのOUの管理権限をほかのCitrix管理者に委任できます。

スクリプトを実行すると、以下の重要なオブジェクトが作成されます。これらのオブジェクトは、標準的なActive Directoryオブジェクトです。スキーマを拡張する必要はありません。

- Controllerのセキュリティグループ。サイト内のすべてのControllerのコンピューターアカウントは、このセキュリティグループに属している必要があります。サイト内のデスクトップは、このセキュリティグループに属しているControllerからのみデータを受け入れます。  
すべてのControllerで、VDAを実行するすべての仮想デスクトップに対する「ネットワーク経由でコンピューターへアクセス」権限が必要です。このため、Controllerのセキュリティグループにこの権限を追加します。Controllerにこの権限がない場合、VDAは登録されません。
- サイトの名前など、サイトに関する情報を含むサービス接続ポイント（SCP）オブジェクト。[Active Directoryユーザーとコンピューター]管理ツールでサイトOUを確認する場合、SCPオブジェクトを表示するには[表示]メニューの[拡張機能]を有効にする必要があります。
- サイトのOU内に作成されるRegistrationServicesという名前のコンテナ。このコンテナには、サイトの各Controllerについて1つのSCPオブジェクトが含まれます。Controllerが起動すると、そのSCPの内容が検証され、必要な場合は更新されます。

初回インストール後に複数の管理者でControllerを追加したり削除したりする必要がある場合は、RegistrationServicesコンテナで子オブジェクトを作成および削除する権限とControllerセキュリティグループでプロパティを書き込む権限が必要です（これらの権限は、Set-ADControllerDiscovery.ps1スクリプトを実行する管理者に自動的に付与されます。）。これらの権限はドメイン管理者または初回インストール時の管理者が付与できます。また、これを行うためのセキュリティグループを設定することをお勧めします。

サイトOUを使用する場合、以下の項目が適用されます。

- Active Directoryに情報が書き込まれるのは、この製品をインストールまたはアンインストールするときや、起動したControllerのSCPの情報を更新するとき（Controllerの名前や通信ポートを変更した後など）のみです。デフォルトでは、Set-ADControllerDiscovery.ps1スクリプトによってサイトOU内のオブジェクトに関する権限が適切にセットアップされ、各ControllerにSCPへの書き込み権限が付与されます。サイトOU内のオブジェクトの内容は、デスクトップとControllerの間の信頼を確立するために使用されます。以下の点を確認してください。
  - 許可された管理者のみが、セキュリティグループのアクセス制御リスト（ACL）を使用してControllerのセキュリティク

ループにコンピューターを追加したり削除したりできること。

- 許可された管理者と各Controllerのみが、ControllerのSCP内の情報を変更できること。
- 展開で複製機能を使用する場合、潜在的な遅延による影響について考慮してください。詳しくは、Microsoftのドキュメントを参照してください。複数のActive Directoryサイトにドメインコントローラーがあるドメイン内でサイトOUを作成する場合は、特に重要になります。デスクトップやController、ドメインコントローラーの場所によっては、サイトOUを最初作成したり、Controllerをインストールまたはアンインストールしたり、Controller名や通信ポートを変更したりしたときにActive Directoryの変更内容が、適切なドメインコントローラーに複製されるまでデスクトップ側に反映されない場合があります。このような遅延が発生するときは、デスクトップがControllerとの接続を確立できないためにユーザーが仮想デスクトップに接続できないなどの問題が発生します。
- この製品では、Active Directoryの標準のコンピューターオブジェクトの属性を複数使用して、デスクトップを管理します。環境によっては、デスクトップのActive Directoryレコードに格納されるコンピューターオブジェクトの完全修飾ドメイン名 (FQDN) が、接続を行うユーザーに返される接続設定の一部として含まれている場合があります。この情報がDN環境の情報と一致していることを確認してください。

## 権限の概要

サイトを作成するには、スクリプトを実行するCitrixサイト管理者が、サイトOUにオブジェクト (SCP、コンテナ、セキュリティグループ) を作成する権限を持っている必要があります。

(サイトOUがない場合、管理者はこれを作成する権限も必要です。ADドメイン管理者は、事前にサイトOUを作成して、Citrixサイト管理者IDに権限を割り当てることをお勧めします。必要な場合、このスクリプトはサイトOUも作成できます。このためには、管理者は、新しいOUの親OUに対する権限を作成する必要があります。ただし、上記のとおり、これはお勧めしません。)

後で、サイトにControllerを追加したり、サイトからControllerを削除したりするには、Citrix管理者にセキュリティグループのマシンを追加/削除する権限、およびSCPを作成/削除する権限が必要です。

通常の操作では、ControllerおよびVDAは、OUおよびその下部で、すべてのオブジェクトに対する読み取り権限が必要です。VDAは、そのマシンIDとしてOUにアクセスします。このマシンIDには、Controllerを検出できるように少なくともOUの読み取り権限が必要です。Controllerには、コンテナで現在のSCPオブジェクトにプロパティを設定する権限が必要です。

Citrix管理者の完全な権限を子OUに付与すると、これらのすべてのアクションが可能になります。ただし、セキュリティ要件が厳格な環境 (スクリプトを実行できるユーザーやアクションの種類を制限しているなど) では、制御の委任ウィザードを使用して特定の権限を設定することができます。次に例を示す手順では、サイトを作成する権限を付与します。

1. 子オブジェクト (サービス接続ポイント (SCP)、コンテナ、セキュリティグループ) を含むOUを作成します。
2. OUを選択して右クリックし、**[制御の委任]** を選択します。
3. 制御の委任ウィザードで、OUの制御を委任するドメインユーザーを指定します。
4. **[委任するタスク]** ページで、**[委任するカスタムタスクを作成する]** を選択します。
5. **[Active Directoryオブジェクトの種類]** ページで、デフォルトの**[このフォルダ、このフォルダ内の既存のオブジェクト、およびこのフォルダ内の新しいオブジェクトの作成]** を受け入れます。
6. **[アクセス許可]** ページで、**[すべての子オブジェクトを作成する権限]** を選択します。
7. ウィザードを終了して、権限を確認します。

「[Controllerの別のサイトへの移動](#)」の手順に従ってください。古いサイトからControllerを削除した後 (手順2)、PowerShellスクリプトSet-ADControllerDiscovery-syncを実行します。

このスクリプトにより、OUが現在のControllerの一覧と同期されます。移動先のサイトに参加した後（手順3.）、そのサイトの任意のController上で同じスクリプトを実行します。

# セッション管理

Jan 04, 2017

最高のユーザーエクスペリエンスを提供するためには、日々のセッションアクティビティを保守することが重要です。ネットワークの信頼性が低い、通信速度が一定していない、ワイヤレスデバイスの伝送距離が制限されているなどの理由でネットワーク接続が失われると、ユーザーの労働意欲が損なわれます。ワークステーション間をすばやく移動でき、ログオンするたびに同じアプリケーションのセットにアクセスできることは、病院の医療スタッフなど多くのモバイルワーカーにとっての優先事項です。

以下の機能を使用すると、セッションの信頼性が最適化され、利便性が向上し、ダウンタイムの増加や生産性の低下を防ぐことができます。また、モバイルユーザーがデバイス間をすばやく移動できるようになります。

- セッション画面の保持
- クライアントの自動再接続
- ICA Keep-Alive
- ワークスペースコントロール

セッション画面の保持機能は、ICAセッションをアクティブのまま保持し、ネットワークの接続が切断されても、セッションの画面を表示したままにできます。ユーザーは、接続が回復するまでセッション画面を見ることができます。

この機能は、ワイヤレス接続を使用するモバイルユーザーにとって特に有用です。たとえば、ワイヤレス接続でのセッション中にトンネルや障害物などの影響で接続に障害が生じた場合、通常はセッションが切断され、セッションの画面が表示されなくなります。この場合、切断セッションに再接続されるまで、そのセッションでは何もできません。セッション画面の保持機能を有効にすると、データを損失することなくセッションがアクティブのまま保持されます。ネットワークが中断されると、セッション画面が停止し、カーソルの形が砂時計に変わるため、ユーザーにもネットワークが切断されていることがわかります。このとき、セッションウィンドウが閉じたりエラーメッセージが表示されたりする代わりに画面表示が保持され、バックグラウンドで再接続が試行されます。ネットワーク接続が回復すると、自動的にセッションでの作業を再開できるようになります。また、セッションに再接続するときに再認証用のログオン画面が表示されないため、ユーザーは即座に作業を再開できます。

Citrix Receiverのユーザーには、Controller側の設定が自動的に適用されます。

セッション画面の保持機能と共にSSL (Secure Sockets Layer) を使用できます。SSLはユーザーデバイスとNetScaler Gateway間で送信されるデータのみを暗号化します。

セッション画面の保持機能は、以下のポリシー設定で構成します。

- [セッション画面の保持] 設定により、セッション画面の保持を許可または禁止します。
- [セッション画面の保持のタイムアウト] 設定では、デフォルトのタイムアウト時間として180秒つまり3分が設定されています。この時間を長く設定することもできますが、この機能の本来の目的は、ネットワークから切断されたユーザーを再認証することなくセッションに再接続することにあるので注意が必要です。必要以上に長い時間を設定すると、接続の再開を待ちきれないユーザーが席を離れてしまい、その間に不正なユーザーがセッションにアクセスしてしまう危険性があります。
- セッション画面の保持機能が有効な受信接続ではポート2598が使用されます。このポート番号はポリシーの[セッション画面の保持のポート番号] 設定で変更できます。
- 切断したセッションに再接続するユーザーを再認証する場合は、クライアントの自動再接続機能を使用します。[クライアントの自動再接続時の認証] 設定を構成して、中断されたセッションにユーザーが再接続するとき再認証を要求することができます。

セッション画面の保持機能とクライアントの自動再接続機能を一緒に使用する場合は、次のように処理されます。まず、

ネットワークが切断されると、セッション画面の保持機能により、セッションがアクティブのままサーバー上に保持されます。[セッション画面の保持のタイムアウト]設定で指定した時間が経過すると、サーバー上のセッションが終了または切断されます。この後でクライアントの自動再接続のポリシー設定が有効になり、切断セッションへの再接続が行われず。

クライアント自動再接続機能では、ネットワークの問題などによって切断されたセッションをReceiverが検出して、そのセッションに自動的に再接続します。この機能がサーバーで有効になっていると、ユーザーは作業を続けるために手動で再接続する必要がありません。

アプリケーションセッションでは、Receiverは、接続に成功するかユーザーがキャンセルするまで再接続を繰り返し試行します。

デスクトップセッションでは、Receiverは、接続に成功するかユーザーがキャンセルするまで、指定された期間の間、再接続を繰り返し試行します。デフォルトでは、この期間は5分です。この期間を変更するには、ユーザーデバイスで以下のレジストリを編集します。

HKLM\Software\Citrix\ICA Client\TransportReconnectRetryMaxTimeSeconds; DWORD;<seconds>

ここで、<seconds>は秒数です。この期間が経過するとセッションへの再接続が試行されなくなります。

クライアント自動再接続機能は、以下のポリシー設定で構成します。

- クライアントの自動再接続。接続が中断した場合のReceiverによる自動再接続を有効または無効にします。
- クライアントの自動再接続時の認証。自動再接続時にユーザーの認証を要求するかどうかを指定します。
- クライアントの自動再接続のログ。再接続イベントのイベントログへの記録を有効または無効にします。ログ機能は、デフォルトで無効になっています。この機能を有効にすると、サーバーのシステムログに自動再接続の成功および失敗イベントが記録されます。再接続イベントのログは、サイト全体で記録されるのではなく、そのイベントが発生した個々のサーバーのシステムログに記録されます。

クライアントの自動再接続機能には、暗号化されたユーザー資格情報に基づく再認証メカニズムが使用されています。ユーザーが最初にログオンしたときに、サーバーにより暗号化されたユーザー資格情報がメモリに格納され、その暗号キーを含んだCookieがクライアント側に送信されます。Receiverは再接続時にこのCookieをサーバーに送信します。サーバーは復号化した資格情報をWindowsのログオンプロセスに送信して認証を求めます。Cookieの有効期限が切れた場合、ユーザーは資格情報を再入力する必要があります。

[クライアントの自動再接続時の認証]設定を有効にする場合、Cookieは使用されません。その代わりに、切断セッションへの再接続時に、ユーザーの資格情報を入力するためのダイアログボックスが開きます。

ユーザーの資格情報とセッションのセキュリティを最大限に保護するために、クライアントとサイトの間すべての通信でSSL暗号化機能を使用してください。

Citrix Receiver for Windowsで自動再接続機能を無効にするには、icaclient.admファイルを編集します。詳しくは、適切なバージョンのCitrix Receiver for Windowsのドキュメントを参照してください。

接続の設定も、クライアントの自動再接続機能に影響します。

- 前述のように、クライアントの自動再接続はポリシー設定のデフォルトによりサイト全体で有効になっています。ユーザーの再認証も不要です。ただし、サーバーでICA TCP接続が切断されたときにセッションをリセットするように設定すると、自動再接続は実行されません。クライアントの自動再接続は、エラーの発生またはタイムアウトによりサーバーがセッションを切断した場合にのみ実行されます。ここでのICA TCP接続とは、実際のネットワーク接続ではなく、TCP/IPネットワーク上のセッションで使用されるサーバーの仮想ポートを指します。
- サーバー上のICA TCP接続では、デフォルトでエラーやタイムアウトが発生した接続のセッションを切断するように設定さ

れています。切断されたセッションはそのままシステムメモリに残るので、ユーザーは同じサーバーに自動的に再接続して、そのセッションでの作業を続行できます。

- エラーが生じたりタイムアウトしたりした接続のセッションについてはリセット、つまりログオフされるように構成することができます。セッションがリセットされた場合、再接続しようとする、切断前の作業状態からセッションが復元されるのではなく、アプリケーションが再起動されて新しいセッションが開始されます。
- セッションがリセットされるようにサーバーが構成されている場合、クライアントの自動再接続により新しいセッションが開始されます。この場合、ユーザーが自分の資格情報を入力して、サーバーにログオンし直す必要があります。
- 外部からの侵入などによってクライアント側から正しくない認証情報が送信された場合、またはセッションの切断が検出されてから自動再接続までの時間が長すぎた場合は、自動再接続に失敗することがあります。

ICA Keep-Alive機能を有効にすると、ネットワークの問題により切断されたセッションにユーザーが再接続できなくなることを防ぐことができます。この機能が有効な場合、セッションのアイドル状態（たとえばクロックデータの更新、マウス操作、画面更新などがない状態）が検出されたときに、リモートデスクトップサービスによりセッションが切断されることを防ぐことができます。サーバーは、定期的にKeep-Aliveパケットを送信して、セッションがアクティブかどうかを検出します。セッションがアクティブでないことが検出されると、サーバーにより「切断」状態として認識されます。

注：ICA Keep-Aliveは、セッション画面の保持機能を使用しない環境でのみ正しく動作します。セッション画面の保持機能では、ICA Keep-Aliveとは異なるメカニズムで切断セッションが管理されます。セッション画面の保持機能を使用しない環境のみ、ICA Keep-Aliveを有効にしてください。

ここでのKeep-Alive機能の設定は、Windowsのグループポリシーによる同様の設定よりも優先されます。

ICA Keep-Alive機能は、以下のポリシー設定で構成します。

- ICA Keep-Aliveタイムアウト。ICA Keep-Aliveメッセージの送信間隔を1~3600秒の範囲で指定します。ただし、ネットワークの問題によるセッションの切断が少なく、アイドル状態のセッションをネットワーク監視ソフトウェアで自動的に閉じるように設定している環境では、このオプションを構成しないでください。  
デフォルト値は60秒で、サーバーからユーザーデバイスにICA Keep-Aliveパケットが60秒おきに送信されます。クライアントが60秒以内に応答しない場合、そのセッションは「切断」状態（タイムアウト）と認識されます。
- ICA Keep-Alive。ICA Keep-Aliveメッセージを送信するかどうかを指定します。

ワークスペースコントロール機能を有効にすると、ユーザーがセッションの途中でデバイスを切り替えても、新しいデバイス上でそのデスクトップやアプリケーションでの作業を継続できます。これにより、ユーザーは自分のデスクトップや作業中のアプリケーションにどこからでもシームレスにアクセスできるようになります。たとえば、病院内の複数のワークステーション間を移動しながら、常に同じアプリケーションセットにアクセスしなければならない医療スタッフをサポートするために、この機能を利用できます。ワークスペースコントロールを構成すると、ユーザーは複数のアプリケーションを一度に切断して、その後で別のクライアントデバイスからそれらのアプリケーションに再接続できます。

ワークスペースコントロールを有効にすると、ユーザーの操作は以下のようになります。

- ログオン：デフォルトでは、ユーザーが移動先でログオンすると、実行されていたすべてのデスクトップおよびアプリケーションに自動的に再接続されます。デスクトップやアプリケーションを手作業で起動する必要はありません。デスクトップまたはアプリケーションのセッションがほかのクライアントデバイス上でアクティブな場合だけでなく、切断されている場合にも接続されます。ユーザーがデスクトップやアプリケーションとの接続を切断しても、サーバー上のセッションは終了しません。管理者は、ユーザーが切断したものだけが再接続されるように構成することもできます。これにより、移動先のクライアントデバイスを使ってユーザーが再ログオンしたときに、前のクライアントデバイスでアクティブなデスクトップやアプリケーションには再接続されず、切断されているものだけが再接続されます。
- 再接続：サーバーに再ログオンしたユーザーは、[再接続] をクリックすることで自分のデスクトップやアプリケーション

ンに一度に再接続できます。デフォルトでは、切断されているデスクトップやアプリケーションと、ほかのクライアントデバイスでアクティブなデスクトップやアプリケーションが再接続されます。管理者は、切断されているデスクトップやアプリケーションだけが再接続されるように構成することもできます。

- **ログオフ**：ユーザーがStoreFront経由でデスクトップやアプリケーションにアクセスする場合に、[ログオフ] コマンドによりStoreFrontおよびすべてのアクティブセッションからログオフするのか、StoreFrontだけからログオフするのかを管理者が構成できます。
- **切断**：ユーザーは、実行中のすべてのデスクトップやアプリケーションを一度に切断できます。個々に切断する必要はありません。

ワークスペースコントロールは、ReceiverユーザーがCitrix StoreFront経由でデスクトップやアプリケーションにアクセスする場合にのみ使用できます。デフォルトでは、仮想デスクトップセッションではワークスペースコントロールが無効になり、ホストされたアプリケーションセッションでは有効になります。公開デスクトップ上で公開アプリケーションを実行する場合、デフォルトではこれらのセッションは共有されません。

ユーザーが別のクライアントデバイスに移動すると、ポリシー、クライアント側ドライブのマッピング、およびプリンター設定が適切に変更されます。ポリシーとクライアント側ドライブのマッピングは、ユーザーがセッションにログオンするクライアントデバイスの条件に基づいて適用されます。たとえば、医療従事者が緊急治療室のクライアントデバイスからログオフして、レントゲン室のワークステーションにログオンして自分のワークスペースに再接続した場合は、レントゲン室でのセッションに適したポリシー、プリンターマッピング、およびクライアント側ドライブのマッピング設定がセッションの開始時に有効になります。

管理者は、ユーザーが場所を移動したときに使用可能になるプリンターをカスタマイズできます。また、ローカルプリンターでの印刷の可否やリモート接続時に使用される帯域幅などの印刷環境を制御することもできます。

ワークスペースコントロール機能を有効にして構成する方法については、StoreFrontのドキュメントを参照してください。

# Studioでの検索の使用

Jan 04, 2017

検索機能を使って、特定のマシン、セッション、マシンカタログ、アプリケーション、またはデリバリーグループに関する情報を表示できます。

1. Studioのナビゲーションペインで[検索]を選択します。

注：[マシンカタログ] ノードや [デリバリーグループ] ノードでは [検索] ボックスを使用できません。ナビゲーションペインの [検索] ノードを使用してください。

追加の検索条件を表示するには、[検索] ボックスの横にあるプラス記号をクリックします。マイナス記号をクリックすると、その検索条件が削除されます。

2. 検索する項目の名前を入力するか、ドロップダウンの一覧からほかの検索オプションを選択します。

3. 検索条件を保存するには、[名前を付けて保存] をクリックします。保存済みの検索は、[保存済みの検索] の一覧に表示されます。

または、矢印アイコン（二重下向き角かっこ）をクリックして検索プロパティの一覧を表示します。この一覧のプロパティで詳細な検索式を定義できます。

高度な検索を行うためのヒント：

- [操作] ペインの [列の選択] を選択すると追加の検索特性を表示させることができ、その特性で検索結果を並べ替えることができます。
- マシンに接続しているユーザーデバイスを検索するには、[クライアント (IP)] および [次のもの] を指定してデバイスのIPアドレスを入力します。
- アクティブなセッションを検索するには、[セッション状態]、[次のもの]、および [接続済み] を指定します。
- 特定のデリバリーグループに含まれるすべてのマシンを一覧で表示するには、ナビゲーションペインで [デリバリーグループ] を選択し、グループを選択して、[操作] ペインの [マシンの表示] を選択します。



# IPv4/IPv6のサポート

Jan 04, 2017

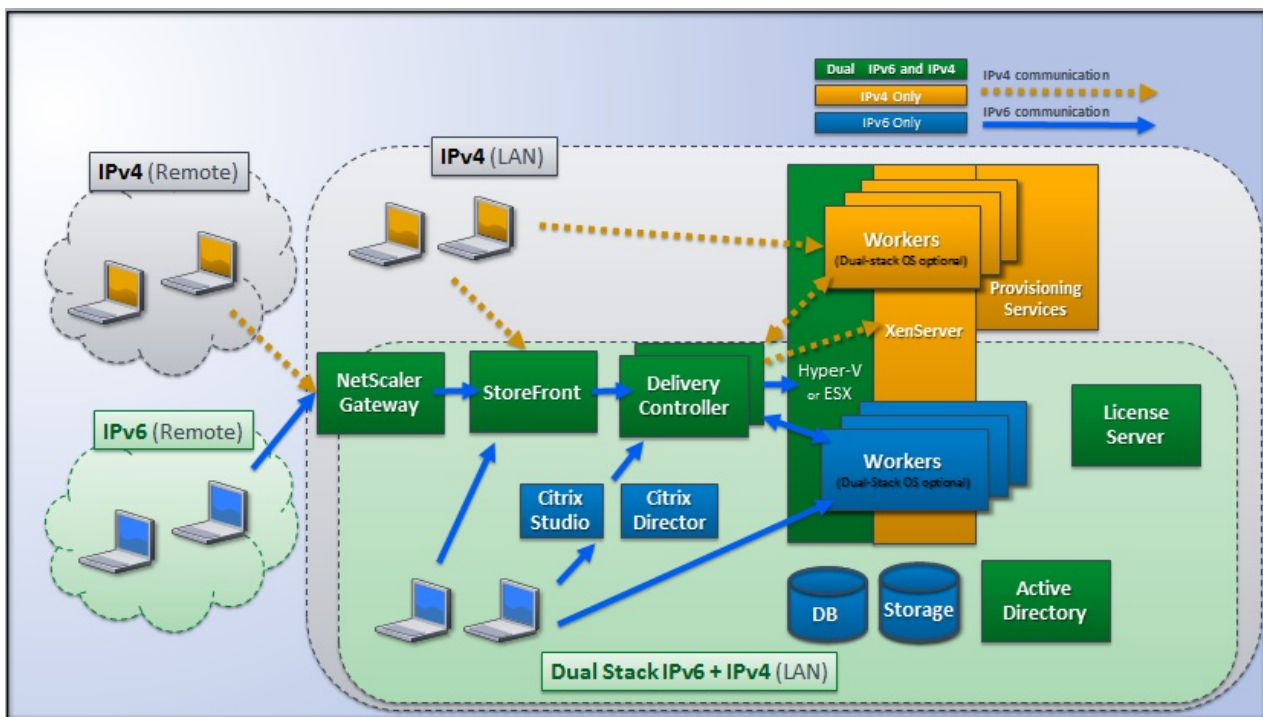
このリリースでは、IPv4のみまたはIPv6のみ（ピュアIPv4またはピュアIPv6）の環境がサポートされ、重複するIPv4とIPv6のネットワークを使用した「デュアルスタック」環境がサポートされます。

IPv6通信は、Virtual Delivery Agent（VDA）接続関連の2つのCitrixポリシー設定で制御されます。

- IPv6を強制的に使用するプライマリ設定：IPv6 Controller登録のみを使用する。
- IPv6ネットマスクを定義する従属設定：コントローラー登録のIPv6ネットマスク。

[IPv6 Controller登録のみを使用する] 設定を有効にすると、VDAはIPv6アドレスで接続を受信するようにDelivery Controllerに登録されます。

次の図は、デュアルスタックIPv4/IPv6展開を示しています。このシナリオで、「ワーカー」とはハイパーバイザーまたは物理システム上にインストールされたVDAを指し、主にアプリケーションやデスクトップへの接続を可能にするために使用されます。デュアルIPv6およびIPv4をサポートするコンポーネントは、トンネリングまたはデュアルプロトコルソフトウェアを使用するオペレーティングシステム上で実行されます。



次のCitrix製品、コンポーネント、および機能ではIPv4のみがサポートされます。

- Provisioning Services
- XenServer 6.x
- [IPv6 Controller登録のみを使用する] ポリシー設定が適用されないVDA
- Version 7.5よりも古いバージョンのXenApp、Version 7よりも古いバージョンのXenDesktop、およびEdgeSight

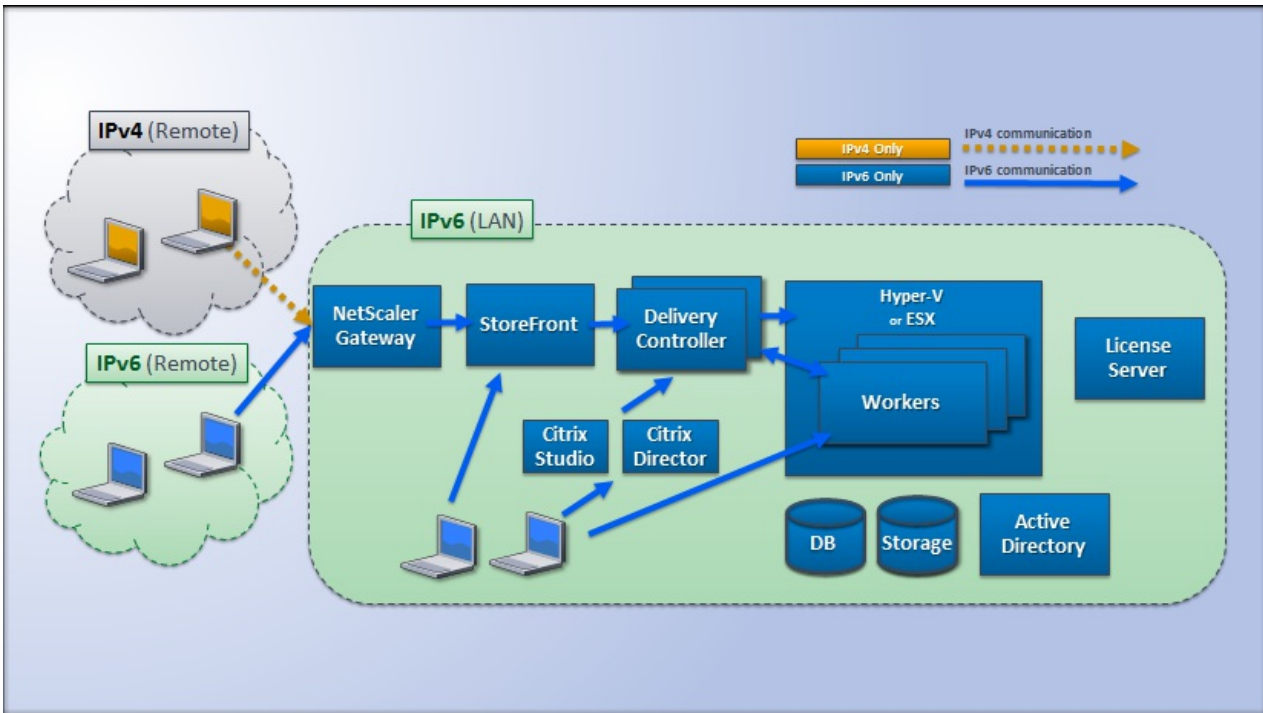
この展開は、以下のように管理されます。

- ユーザーによるIPv6ネットワークの使用頻度が高く、管理者がユーザーにIPv6トラフィックの使用を求める場合、管理者にプライマリのIPv6ポリシー設定（つまり [IPv6 Controller登録のみを使用する]）が有効な組織単位（OU）またはワーカーイメージを使用してIPv6デスクトップとアプリケーションを公開します。

- ユーザーによるIPv4ネットワークの使用頻度が高い場合、管理者はプライマリのIPv6ポリシー設定（つまり [IPv6 Controller登録のみを使用する]）が無効（デフォルト）なOUまたはワーカーイメージを使用してIPv4デスクトップとアプリケーションを公開します。

次の図は、ピュアIPv6展開を示しています。このシナリオの内容は以下のとおりです。

- 各コンポーネントは、IPv6ネットワークのサポートが構成されたオペレーティングシステム上で実行されている。
- すべてのVDAに対してプライマリのIPv6ポリシー設定（ [IPv6 Controller登録のみを使用する] ）が有効になっている。つまり、VDAをIPv6アドレスを使ってControllerに登録する必要がある。



Citrixポリシーには、ピュアIPv6環境やIPv4/IPv6デュアルスタック環境をサポートするための2つの設定項目があります。次の接続関連ポリシーを構成します。

- IPv6 Controller登録のみを使用する — Delivery Controller登録でVirtual Delivery Agent (VDA) により使用されるアドレスの形式を制御します。デフォルトでは、無効に設定されています。
  - VDAがControllerと通信を行うときに、グローバルIPアドレス、ユニークローカルアドレス (ULA)、リンクローカルアドレス（ほかのIPv6アドレスを使用できない場合のみ）の順で単一のIPv6アドレスが選択されます。
  - この設定項目が無効な場合、そのマシンのIPv4アドレスを使用してVDAがControllerと登録および通信を行います。
- コントローラー登録のIPv6ネットマスク — 1つのマシンが複数のIPv6アドレスを保持することがあります。この設定項目では、VDAで使用されるサブネットを指定できます。この場合、グローバルIPは使用されません。これにより、VDAが登録するネットワークが、指定されたネットマスクに最初にマッチしたアドレスのみに限定されます。この設定項目を使用する場合は、 [IPv6 Controller登録のみを使用する] 設定を有効にする必要があります。デフォルトでは空文字が設定されています。

重要：これらの設定項目によってのみ、VDAで使用されるアドレスの形式 (IPv4またはIPv6) が決定されます。つまり、VDAでIPv6アドレスが使用されるようにするには、 [IPv6 Controller登録のみを使用する] 設定を有効にしたポリシーを適用する必要があります。

環境内にIPv4とIPv6の両方のネットワークがある場合、IPv4のみのクライアントとIPv6ネットワークにアクセスできるクライアントに対して、別個のデリバリーグループ構成が必要です。ユーザーを区別するために、名前付け、手動Active Directoryグループ割り当て、またはSmartAccessフィルターの使用を検討してください。

IPv6ネットワークで接続されたセッションにIPv4アクセスのみの内部クライアントから再接続する場合、再接続に失敗することがあります。

# クライアントフォルダーのリダイレクト

Jan 04, 2017

クライアントフォルダーのリダイレクトは、クライアント側のファイルがホスト側のセッションのへアクセスする方法を変更します。ホスト側でクライアント側ドライブのマッピングのみを有効にすると、クライアントの側の全ボリュームがUNC (Universal Naming Convention) リンクとしてセッションに自動的にマップされます。管理者がホスト側でクライアントフォルダーのリダイレクトを有効にして、ユーザーがそれをユーザーデバイス上で構成すると、ユーザーが指定したローカルボリュームの一部がリダイレクトされます。

セッション内では、ユーザーデバイス上のすべてのファイルシステムの代わりにユーザー指定のフォルダーのみがUNCリンクとして表示されます。レジストリでUNCリンクを無効にすると、クライアントフォルダーはマップされたドライブとしてセッション内で表示されます。

クライアントフォルダーのリダイレクトはWindowsデスクトップOSマシンでのみサポートされます。

外部USBドライブに対するクライアントフォルダーのリダイレクトは、デバイスを解除して再接続しても保存されません。

ホスト側でクライアントフォルダーのリダイレクトを有効にします。次に、クライアントデバイス上でリダイレクト対象フォルダーを指定します。クライアントフォルダーオプションの指定に使用するアプリケーションは、このリリースで提供されるCitrix Receiverに含まれています。

注意：レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

## 1. ホスト側：

1. キー (HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\Client Folder Redirection) を作成します。
2. REG\_DWORD値を作成します。
  - 値の名前：CFROnlyModeAvailable
  - 種類：REG\_DWORD
  - 値のデータ：1

## 2. ユーザーデバイス側で以下を行います。

1. 最新バージョンのReceiverがインストールされていることを確認します。
2. Receiverのインストール先ディレクトリで、CtxCFRUI.exeを実行します。
3. [Custom] ラジオボタンをクリックし、フォルダーを追加、編集、または削除します。
4. セッションを切断してから再接続すると、変更が適用されます。

# Personal vDisk 7.x

Jan 04, 2017

Personal vDisk機能を使用すると、プールされるデスクトップやストリーム配信されるデスクトップを単一のイメージで管理でき、しかもユーザーによるアプリケーションのインストールやデスクトップ設定の変更が可能になります。従来の仮想デスクトップインフラストラクチャ (VDI) では、仮想デスクトップでユーザーが設定を変更したりアプリケーションをインストールしたりしても、管理者がマスターイメージを更新するたびにそれらの変更が破棄されてしまいます。Personal vDiskを使用すると、ユーザーによる変更がそのまま保持されます。管理者は、ユーザーによるデスクトップのカスタマイズや個人設定を許可しながら、マスターイメージを容易に一元管理できます。

Personal vDiskでは、ユーザーの仮想マシンに対するすべての変更をその仮想マシンに割り当てられた別ディスク (Personal vDisk) にリダイレクトすることにより、変更内容を保持します。Personal vDiskに保存された内容はデスクトップの実行時にマスターイメージの内容と統合され、ユーザーに一貫した操作環境が提供されます。この方法では、管理者がマスターイメージでプロビジョニングしたアプリケーションにもユーザーは引き続きアクセスできます。

Personal vDiskは、デフォルトで同じ容量の2つの領域で構成されます。

- ユーザープロファイル：ここには、ユーザーデータ、ドキュメント、およびユーザープロファイルが格納されます。デフォルトではドライブ文字「P」が割り当てられますが、マシンカタログを作成するときに別のドライブ文字を選択することもできます。使用されるドライブの設定は、レジストリキーEnableUserProfileRedirectionにも依存します。
- 仮想ハードディスク (VHD) ファイル：ここには、そのほかのすべてのファイル (C:\Program Filesにインストールされたアプリケーションなど) が格納されます。この部分はWindows Explorerには表示されず、Version 5.6.7以降ではドライブ文字は必要ありません。

Personal vDiskでは、個々のユーザーがダウンロードしてインストールするアプリケーションに加えて、部門レベルでプロビジョニングするアプリケーションがサポートされます。これにはドライバー (Phase 1ドライバーを除く) やデータベースを必要とするアプリケーション、およびマシン管理ソフトウェアなどがあります。ユーザーによる変更と管理者による変更が併合する場合でも、Personal vDiskの機能を使って簡単かつ自動的に解決できます。

さらに、ローカルで管理されるアプリケーション (ローカルのIT部門によりプロビジョニングされて管理されるアプリケーションなど) をユーザーの環境にプロビジョニングすることもできます。Personal vDiskを使用する場合でも、ユーザーの操作性は変更されません。ユーザーが変更した設定やインストールしたアプリケーションは、自動的にPersonal vDisk上に格納されます。Personal vDisk上のアプリケーションがマスターイメージ上のものとまったく同じである場合、Personal vDisk上のアプリケーションが破棄されます。これにより、そのアプリケーションは使用可能なままPersonal vDiskの容量が節約されません。

物理的には、Personal vDiskをハイパーバイザーに格納します。ただし、仮想デスクトップにアタッチされているほかのディスクと同じ場所に配置する必要はありません。これにより、Personal vDiskストレージのコストを削減できます。

サイトの作成中、コネクションを作成するときに、仮想マシンで使用されるディスクのストレージを指定します。Personal vDiskは、オペレーティングシステム用のディスクとは異なるストレージに配置できます。各仮想マシンは、どちらのストレージにもアクセスできる必要があります。これらのディスクをローカルストレージに配置する場合は、同じハイパーバイザーからアクセスできる必要があります。このため、Studioにはこれらの条件を満たすストレージのみが表示されます。後で、Studioで [構成] の [ホスト] を選択して、Personal vDiskとそのストレージを既存のホストに追加することもできます (マシンカタログには追加できません)。

Personal vDiskは、適切な方法で定期的にバックアップしてください。vDiskはハイパーバイザーのストレージ層の標準のボリュームであるため、ほかのボリュームと同様の方法でバックアップできます。

このリリースには、以下の改善が含まれています。

- Personal vDiskのパフォーマンスが向上し、Personal vDiskカタログにイメージ更新を適用するときの時間が短くなりました。

このリリースでは、以下の問題が解決されています。

- 基本仮想マシン上のMicrosoft Office 2010をMicrosoft Office 2013にインプレースアップグレードすると、再構成のウィンドウが開いた後で「エラー25004。入力したプロダクトキーは、このコンピューターでは使用できません」というエラーメッセージが表示されることがありました。これまでのリリースでは、基本仮想マシンからOffice 2010をアンインストールしてからOffice 2013をインストールすることでこのエラーを避けることができました。このリリースでは、Office 2010をアンインストールせずにインプレースアップグレードできます (#391225)。
- ユーザーのPersonal vDisk上に新しいバージョンのMicrosoft .Netが存在する場合、イメージの更新時に基本仮想マシン上の以前のバージョンで上書きされるという問題がありました。この問題により、新しいバージョンの.Netを使用するPersonal vDisk上のアプリケーション (Visual Studioなど) が正しく動作しなくなることがありました (#439009)。
- Personal vDiskが有効なProvisioning ServicesイメージディスクでPersonal vDiskを使用しないマシンカタログを作成できないという制限事項がありました。このリリースでは、この制限事項はありません (#485189)。

Version 7.6には、以下の新機能が追加されています。

- 拡張されたPersonal vDiskのエラー処理およびレポート。Personal vDiskが有効なマシンをStudioのマシンカタログで表示すると、[PvD] タブにイメージ更新時の監視状態や推定完了時間、および進行状況が表示されます。また、より詳細な状態情報が表示されるようになりました。
- 以前のバージョンで提供されていたPersonal vDiskイメージ更新の監視ツール (Image Update Monitoring Tool) が、ISOイメージのISO\Support\Tools\Scripts\PvdToolフォルダーに収録されています。以前のバージョンでも監視機能が提供されていましたが、このリリースではより高度なレポート機能が提供されます。
- Provisioning Servicesのテストモジュールを使用すると、テストカタログ内の更新済みイメージでマシンを起動できます。起動したマシンの安定性を確認した後で、テスト用のPersonal vDiskを実稼働用に昇格させることができます。
- インベントリ実行時に、(Personal vDiskデスクトップ単位ではなく) 2つのインベントリの相違を計算できるようになりました。新しく追加されたコマンドを使用して、MCSカタログの既存のインベントリをエクスポートしてインポートできます。(Provisioning ServicesのマスターvDiskの場合は、以前のインベントリが既に含まれています。)

Version 7.1.3で確認されVersion 7.6で解決された問題は以下のとおりです。

- Personal vDiskのアップグレードを中断すると、既存のPersonal vDiskが破損することがありました。[#424878]
- Personal vDiskを長時間実行して非ページメモリリークが発生すると、仮想デスクトップが応答不能になることがありました。[#473170]

Version 7.6で新たに確認された問題は以下のとおりです。

- アンチウイルス製品をインストールすると、インベントリや更新に時間がかかる場合があります。CtxPvD.exeおよびCtxPvDSvc.exeをアンチウイルス製品の除外プロセスの一覧に追加すると、パフォーマンスを向上させることができます。これらのファイルは、C:\Program Files\Citrix\personal vDisk\binにあります。[#326735]
- マスターイメージから継承されたファイル間のハードリンクは、Personal vDiskカタログで保持されません。[#368678]
- Personal vDiskマスターイメージ上のOffice 2010をOffice 2013にアップグレードすると、仮想マシン上でのOfficeの起動に失敗することがあります。これは、アップグレード時にOffice KMSライセンスプロダクトキーが削除されるために発生します。この問題を回避するには、マスターイメージ上でOffice 2010をアンインストールしてからOffice 2013を再インストールしてください。[#391225]
- Personal vDiskカタログはVMware Paravirtual SCSI (PVSCSI) コントローラーをサポートしません。デフォルトのコントローラーを使用してください。[#394039]
- Personal vDisk Version 5.6.0を使用する仮想デスクトップでユーザーが作成したファイルが、Personal vDisk Version 7への

アップグレード後に消失することがあります。この問題は、プールされた仮想マシンにログオンしたときに新しいユーザープロファイルが作成されるために発生します。現在、この問題を回避する方法はありません。[#392459]

- Windows 7を実行するPersonal vDiskでWindowsシステム保護機能が有効な場合、バックアップと復元機能を使用できません。システム保護を無効にするとユーザープロファイルがバックアップされますが、userdata.v2.vhdファイルは除外されます。システム保護を無効にしてバックアップと復元機能を使ってユーザープロファイルのバックアップを作成することをお勧めします。[#360582]
- ディスクの管理ツールを使用して基本仮想マシン上でVHDファイルを作成する場合、VHDをマウントできないことがあります。この問題を解決するには、Personal vDiskボリュームにVHDをコピーしてください。[#355576]
- Office 2010ソフトウェアを削除しても、そのショートカットが仮想デスクトップから削除されません。ショートカットは手作業で削除してください。[#402889]
- Microsoft Hyper-V環境でPersonal vDiskを使用するマシンカタログを作成するときに、ローカルに格納されたマシンとクラスターの共有ボリューム (CSV) に格納されたvDiskを指定すると、エラーが発生して処理に失敗します。この問題を回避するには、vDiskにほかのストレージを使用してください。[#423969]
- Provisioning Servicesを使用するマシンカタログの仮想デスクトップに初めてログオンする場合、(ctxpvd.exe -s resetコマンドで) Personal vDiskがリセットされていると、デスクトップの再起動を求めるメッセージが表示されます。この場合、デスクトップを再起動する必要があります。この問題は初回ログオン時にのみ発生し、それ以降のログオン時に再起動は要です。[#340186]
- .NET 4.5をPersonal vDiskにインストールした後で、イメージの更新により.NET 4.0をインストールまたは修正すると、.NET 4.5に依存するアプリケーションを実行できなくなります。この問題を回避するには、基本イメージに.NET 4.5をインストールしてイメージを更新してください。
- XenApp/XenDesktop 7.6の「  
— 既知の問題  
」のトピックも参照してください。

Version 7.1.1で確認されVersion 7.1.3で解決された問題は以下のとおりです。

- Personal vDisk 5.6.0からPersonal vDisk 7.xへの直接アップグレードにより、Personal vDiskに問題が発生することがありました。[#432992]
- Personal vDiskを使用する仮想デスクトップにユーザーが断続的に接続できなくなる場合があります。[#437203]
- Personal vDisk 5.6.5以降からPersonal vDisk 7.0以降へのアップグレード時にPersonal vDiskのイメージ更新処理が中断されると、それ以降の更新処理に失敗することがありました。[#436145]

Version 7.1で確認されVersion 7.1.1で解決された問題は以下のとおりです。

- イメージ更新を介してSymantec Endpoint Protection 12.1.3を更新すると、symhelp.exeによりアンチウイルス定義の破損が報告されます。[#423429]
- サービスコントロールマネージャー (services.exe) がクラッシュすると、プールされたデスクトップがPersonal vDiskにより再起動されることがありました。[#0365351]

Version 7.1.1で新たに確認された問題はありません。

Version 7.1には、以下の新機能が追加されています。

- Windows 8.1を実行するデスクトップを備えたPersonal vDiskを使用できるようになりました。また、イベントログ機能も向上しています。
- このリリースでは、コピーオンライト (CoW) はサポートされなくなりました。Personal vDiskをバージョン7.0から7.1にアップグレードすると、CoWによって管理されたデータに対するすべての変更が失われます。これはXenDesktop 7の試験

的機能であり、デフォルトでは無効になっていました。そのため、この機能を有効にしなかった場合には、特に影響を受けません。

Version 7.0.1で確認されVersion 7.1で解決された問題は以下のとおりです。

- Personal vDiskレジストリキーのEnableProfileRedirectionの値を1またはONに設定し、また後からイメージの更新中に値を0またはOFFに変更すると、Personal vDiskスペース全体がユーザーによってインストールされたアプリケーションに割り当てられ、ユーザープロファイル用の領域がなくなってvDisk上に残ることがありました。このプロファイルのリダイレクトがカタログに対して無効で、イメージの更新中にそれを有効にする場合、ユーザーは仮想デスクトップにログオンすることができないことがあります。[#381921]
- Personal vDiskのインベントリ更新に失敗すると、デスクトップサービスによって正確なエラーがイベントビューアーに記録されませんでした。[#383331]
- Personal vDisk 7.xにアップグレードするときに、変更済の規則が保持されないという問題がありました。この問題は、Version 7.0からVersion 7.1へのアップグレードでは発生しません。ただし、Version 5.6.5をVersion 7.1にアップグレードする場合は、最初に規則ファイルを保存し、次にアップグレードの後で規則を再度適用する必要があります。[#388664]
- Windows 8が動作するPersonal vDiskで、Windowsストアからアプリケーションをインストールできないという問題がありました。この場合、「購入処理は完了できませんでした」という内容のエラーメッセージが表示されます。Windows Updateサービスを有効にしてもこの問題は解決しませんでした。現在はこの問題が解決されています。ただし、ユーザーがインストールしたアプリケーションは、システムの再起動後に再インストールする必要があります。[#361513]
- Personal vDiskを使用するWindows 7のプールされたデスクトップで、一部のシンボリックリンクが欠落するという問題がありました。この問題により、C:\Users\All Usersにアイコンを格納するアプリケーションでは、[スタート]メニューにアイコンが表示されませんでした。[#418710]
- インベントリの更新後にシステムに多くの変更を追加すると、更新シーケンス番号 (USN) ジャーナルオーバーフローが発生し、Personal vDiskが起動しないという問題がありました。[#369846]
- 状態コード0x20およびエラーコード0x20000028により、Personal vDiskが起動しませんでした。[#393627]
- Symantec Endpoint Protection 12.1.3により「Proactive Threat Protectionは誤動作しています」というメッセージが表示され、このコンポーネントのLive Update Statusを使用できませんでした。[#390204]

Version 7.1で新たに確認された問題については、XenDesktop 7.1の「

— 既知の問題

」のトピックを参照してください。

Version 7.0.1の新機能：環境の変更に対してPersonal vDiskがより強固になりました。イメージ更新に失敗した場合でもPersonal vDiskを伴う仮想デスクトップをDelivery Controllerに登録できるようになっており、また安全でないシステムシャットダウンによりvDiskが使用不可となってしまうことがなくなりました。さらに、規則ファイルを使用することで、展開時にvDiskからファイルやフォルダーを除外できるようになりました。

Version 5.6.13で確認されVersion 7.0.1で解決された問題は以下のとおりです。

- プールされた仮想デスクトップ上でのユーザーによるグループのメンバーシップの変更内容が、イメージの更新後に失われることがありました。[#286227]
- Personal vDiskに十分な領域がある場合でも、イメージ更新がディスク容量不足によるエラーで失敗することがありました。[#325125]
- 一部のアプリケーションで仮想デスクトップのPersonal vDiskへのインストールに失敗し、再起動を求めるメッセージが表示されることがありました。これは、保留中の名前変更操作によるものです。[#351520]
- マスターイメージ内で作成されたシンボリックリンクが、Personal vDiskを使用する仮想デスクトップで正しく機能しないという問題がありました。[#352585]
- Citrix Profile managementとPersonal vDiskを使用する環境では、プロファイルのリダイレクトが有効な場合にシステムボリューム上でユーザープロファイルを調査するアプリケーションは正常に機能しないことがありました。[#353661]



- インベントリのサイズが2GBを超える場合に、マスターイメージ上でのインベントリの更新に失敗するという問題がありました。[#359768]
  - イメージの更新がエラーコード112により失敗し、Personal vDiskが破損することがありました。この問題は、更新に必要な空き領域がvDiskにある場合でも発生します。[#363003]
  - カタログのデスクトップ数が250を超える場合に、サイズ変更スクリプトの処理が失敗するという問題がありました。[#363365]
  - ユーザーによる環境変数に対する変更が、イメージの更新時に失われることがありました。[#372295]
  - Personal vDiskを使用する仮想デスクトップ上で作成されたローカルユーザーが、イメージ更新時に失われるという問題がありました。[#377964]
  - インベントリの更新後にシステムに多くの変更を追加すると、更新シーケンス番号 (USN) ジャーナルオーバーフローが発生し、Personal vDiskの起動に失敗するという問題がありました。この問題を避けるには、USNジャーナルのサイズをマスターイメージ内で32 MB以上に増やして、イメージ更新を実行します。[#369846]
  - Personal vDiskでAppSense Environment Managerを置き換えモードで使用する場合、レジストリハイブが正しく処理されないという問題が確認されています。これは、Personal vDiskがインストールされる時のRegRestoreKey APIの動作に関連しています。CitrixとAppSenseは共同でこの問題の解決にあたっています。[#0353936]
- Personal vDisk上にインストールされたアプリケーションがマスターイメージ上にインストールされた同一バージョンのアプリケーションと関連付けられている場合、イメージの更新後にPersonal vDisk上のアプリケーションが動作しなくなることがあります。この問題は、マスターイメージ上のアプリケーションをアンインストールしたりアップグレードしたりしたために、Personal vDisk上のアプリケーションに必要なファイルがマスターイメージから削除されると発生します。この問題を回避するには、マスターイメージ上のアプリケーションを保持しておきます。
- たとえば、マスターイメージ上にOffice 2007をインストールして、Personal vDisk上にVisio 2007をインストールします。その後で管理者がマスターイメージ上のOffice 2007をOffice 2010にアップグレードして、そのイメージを使用してマシンを更新すると、Visio 2007が動作しなくなります。この問題を避けるには、Office 2007をマスターイメージ上にインストールしたままにしておきます。[#320915]
- Personal vDiskを使用するデスクトップにMcAfee Virus Scan Enterprise (VSE) を展開する場合は、マスターイメージ上にVersion 8.8 Patch 4以降をインストールしてください。[#303472]
  - マスターイメージ内で作成されたファイルのショートカットが動作しなくなった場合（そのショートカットのリンク先の名前がPersonal vDisk内で変更された場合など）は、ショートカットを作成し直してください。[#367602]
  - マスターイメージ内で絶対/ハードリンクを使用しないでください。[#368678]
  - Windows 7のバックアップと復元機能は、Personal vDiskでサポートされていません。[#360582]
  - 更新したマスターイメージを適用した後で、ローカルユーザーおよびグループのコンソールがアクセス不能になったり不十分なデータが表示されたりすることがあります。この問題を解決するには、仮想マシン上でユーザーアカウントをリセットします。これを行うには、セキュリティハイブのリセットが必要です。この問題はVersion 7.1.2で解決されており、Version 7.1.2以降で作成された仮想マシンは正しく処理されます。ただし、アップグレード前に作成された仮想マシンでの問題は解決されません。[#488044]
  - ESXハイパーバイザー環境でプールされた仮想マシンを使用する場合、SCSIコントローラーの種類として [VMware Paravirtual] を選択すると再起動を確認するメッセージがユーザーに表示されます。この問題を避けるには、SCSIコントローラーの種類としてLSIを使用してください。[#394039]
  - Provisioning Servicesで作成されたデスクトップ上でPersonal vDiskをリセットすると、ログオンしたユーザーに再起動を確認するメッセージが表示されることがあります。この問題が発生した場合は、デスクトップを再起動してください。[#340186]
  - Windows 8.1のデスクトップを使用するユーザーがPersonal vDiskにログオンできなくなることがあります。この問題が発生すると、「システムが正しくシャットダウンされなかったためPersonal vDiskが無効になっています」というメッセージが管理者に表示され、PvDActivation.logに「Failed to load reg hive

[\Device\IvmVhdDisk00000001\CitrixPvD\Settings\RingCube.dat]』というメッセージが記録されます。この問題は、ユーザーの仮想マシンを正しくシャットダウンできなかった場合に発生します。この問題を解決するには、Personal vDiskをリセットしてください。[#474071]

# インストールとアップグレード

Jan 04, 2017

Personal vDisk 7.xは、XenDesktop Version 5.6以降でサポートされています。XenDesktopの各バージョンの「システム要件」のトピックを参照して、サポートされるVirtual Delivery Agent (VDA) のオペレーティングシステム、ホスト（仮想化リソース）、およびProvisioning Servicesについて確認してください。Provisioning Servicesの操作について詳しくは、Provisioning Servicesのドキュメントを参照してください。

マシン上のVDA for Desktop OSをインストールしたりアップグレードしたりするときに、Personal vDiskも自動的にインストールされます。VDAのインストール後にPersonal vDiskをアップデートする場合は、[ここで提供されているPersonal vDiskのMSIファイル](#)を使用してください（アカウントの資格情報が必要です）。

Personal vDiskは、以下の状況で有効になります。

- Machine Creation Services (MCS) を使用している場合、Personal vDisk用のデスクトップOSマシンカタログの作成時にPersonal vDiskが自動的に有効になります。
- Provisioning Services (PVS) を使用している場合、マスター（基本）イメージ作成時に管理者がインベントリを実行したとき、および自動更新によるインベントリ実行時にPersonal vDiskが自動的に有効になります。

VDAをインストールするときに、Personal vDiskを有効にするためのオプション（インストールウィザードの [Personal vDisk] チェックボックスまたはコマンドラインインターフェイスの/baseimageオプション）を選択できます。これらのオプションを選択せずに（つまりデフォルトのまま）VDAをインストールした場合でも、カタログ作成時にPersonal vDiskが有効になるため、同じイメージを使用してPersonal vDiskが有効なデスクトップおよびPersonal vDiskが無効なデスクトップを作成できます。

新しいサイトを構成するときに、Personal vDiskをホストに追加します。ホスト上の同じストレージを仮想マシンとPersonal vDisk用に使用したり、Personal vDisk用に専用のストレージを使用したりできます。

その後で、Personal vDiskとそのストレージを既存のホスト（接続）に追加することもできます（マシンカタログには追加できません）。

1. Studioのナビゲーションペインで [構成] 、 [ホスト] の順に選択します。
2. [操作] ペインの [Personal vDiskストレージの追加] を選択し、ストレージの場所を指定します。

Personal vDisk Version 7.xを簡単にアップグレードするには、VDA for Desktop OSを最新のXenDesktopレベルにアップグレードします。その後で、Personal vDiskのインベントリを実行します。

Personal vDiskのみをアップグレードするには、[ここで提供されているPersonal vDiskのMSIファイル](#)を使用します。

Personal vDiskソフトウェアをアンインストールするには、以下のいずれかの方法を使用します。

- VDAをアンインストールします。これにより、Personal vDiskソフトウェアもアンインストールされます。
- Personal vDiskのMSIファイルを使用してPersonal vDiskをアップデートした場合は、コントロールパネルを使用してアンインストールできます。

Personal vDiskをアンインストールして同じまたは新しいバージョンを再インストールする場合は、事前にレジストリキー

HKEY\_LOCAL\_MACHINE\Software\Citrix\personal vDisk\configをバックアップしておいてください。このレジストリキーには、変更された環境構成設定が含まれています。Personal vDiskを再インストールしたら、バックアップしたレジストリ情報を使用して、必要に応じて値を変更してください。

# 構成と管理

Jan 04, 2017

このトピックでは、Personal vDisk (PVD) 環境を構成したり管理したりするときに考慮すべき内容について説明します。また、推奨される構成やタスクについても説明します。

Windowsレジストリの編集が必要な操作を行う場合の注意事項。

注意：レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

メインのPersonal vDiskのサイズを決定するときは、以下の要因について考慮します。

- **ユーザーがPersonal vDisk上にインストールするアプリケーションのサイズ**

Personal vDiskでは、再起動時にアプリケーション領域 (UserData.v2.vhd) の空き容量が確認されます。空き容量が全体の10%未満になると、未使用のプロファイル領域 (デフォルトではPドライブの使用可能領域) を使ってアプリケーション領域が拡張されます。このときアプリケーション領域に追加されるスペースは、アプリケーション領域とプロファイル領域で使用可能な合計空き容量の約50%です。

たとえば、10GBのPersonal vDisk上のアプリケーション領域が5GBでプロファイル領域の空き容量が3GBの場合、アプリケーション領域の4.7GBが消費されたときに自動的に追加されるスペースは以下の式で算出されます。

$$\text{追加されるスペース} = (5.0 - 4.7) \div 2 + 3.0 \div 2 = 1.65\text{GB}$$

アプリケーション領域に追加されるスペースのサイズは概算値です。これは、ログやオーバーヘッド用にわずかな余分スペースが追加されるためです。この計算およびサイズ調整は、各再起動時に行われます。

- **ユーザープロファイルのサイズ (ほかのプロファイル管理ツールを使用しない場合)**

アプリケーションに必要な容量に加えて、ユーザープロファイルの格納に十分な容量がPersonal vDisk上にあることを確認してください。また、リダイレクトされないユーザーフォルダー (マイドキュメントやマイミュージックなど) の容量についても考慮する必要があります。既存のプロファイルのサイズを確認するには、コントロールパネルから [システムのロパティ] (sysdm.cpl) を開きます。

プロファイルをリダイレクトする一部のツールを使用すると、実際のプロファイルデータの代わりにスタブファイル (センチネルファイル) が格納されます。これらのプロファイル管理ツールでは初期状態でディスクが消費されていないように見えますが、各スタブファイルについて1ファイルディレクトリエントリがファイルシステム上に作成されます (通常、各ファイルについて4KB程度)。このようなプロファイル管理ツールを使用する場合は、スタブファイルではなく実際のプロファイルデータのサイズを考慮する必要があります。

エンタープライズクラスのファイル共有アプリケーション (ShareFile、Dropboxなど) により、Personal vDisk上のユーザープロファイル領域がデータの同期に使用される場合があります。このようなアプリケーションを使用する場合は、同期データのサイズも考慮する必要があります。

- **Personal vDiskインベントリを含んでいるテンプレートVHDで使用される容量**

テンプレートVHDには、Personal vDiskインベントリデータ (マスターイメージの内容に対応するセンチネルファイル) が含まれています。Personal vDiskのアプリケーション領域はこのVHDから作成されます。各センチネルファイルやフォルダーによりファイルディレクトリエントリが構成されるため、ユーザーがアプリケーションをインストールしていなくてもPersonal vDiskのアプリケーション領域がテンプレートVHDの内容により消費されます。テンプレートVHDのサイズは、

インベントリを実行した後でマスターイメージを参照すると確認できます。または、以下の式でおおよそのサイズを算出します。

テンプレートVHDのサイズ = マスターイメージ上のファイル数 × 4KB

マスターイメージ上のファイルおよびフォルダーの数を確認するには、そのイメージのCドライブを右クリックして[プロパティ]を選択します。たとえば、イメージのファイル数が250,000の場合、テンプレートVHDのサイズは約1,024,000,000バイト（1GB弱）になります。つまり、Personal vDiskのアプリケーション領域のうち1GB弱のディスクスペースには、アプリケーションをインストールできません。

#### ● Personal vDiskイメージの更新時に使用される容量

Personal vDiskイメージを更新するときには、イメージの2つのバージョンの差分やユーザーによるPersonal vDiskの変更内容を統合するために十分な空き領域が、Personal vDisk（デフォルトでPドライブ）のルートに必要になります。通常、Personal vDiskの200～300MBがこの目的で予約されます。ただし、Pドライブに追加されるデータの量によっては、イメージの更新に必要な容量を確保できなくなることがあります。Personal vDiskプール統計スクリプト（XenDesktopインストールメディアのSupport/Tools/Scriptsフォルダー）またはPersonal vDiskイメージ更新監視ツール

（Support/Tools/Scripts\PvdToolフォルダー）を使用して、更新対象のカatalogから空き領域が少ないPersonal vDiskディスクを特定できます。

アンチウイルス製品をインストールすると、インベントリや更新に時間がかかる場合があります。CtxPvD.exeおよびCtxPvDSvc.exeをアンチウイルス製品の除外の一覧に追加すると、パフォーマンスを向上させることができます。これらのファイルは、C:\Program Files\Citrix\personal vDisk\binにあります。ウイルスチェックの対象からこれらの実行可能ファイルを除外すると、インベントリおよびイメージ更新の処理パフォーマンスが最大で10倍に向上することがあります。

#### ● 予定外の追加容量（計画外のアプリケーションのインストールなど）

ユーザーが追加のアプリケーションをインストールできるように、（特定サイズまたは全体に対する割合で）追加領域を計算することを検討してください。

管理者は、Pドライブに対するVHDの相対的なサイズを決定するときの自動サイズ変更アルゴリズムを手動で調整できます。これを行うには、VHDの初期サイズを設定します。たとえば、ユーザーがインストールするアプリケーションの数が多いために、デフォルトのアルゴリズムで決定されるサイズでは足りなくなることがわかっている場合などには、この機能が役に立ちます。この場合、アプリケーションをインストールするための領域が足りなくならないように、アプリケーション領域の初期サイズを増やします。

可能な場合は、マスターイメージ上でVHDの初期サイズを調整します。または、仮想デスクトップ上でVHDのサイズを調整して、ユーザーのアプリケーションのインストールに必要な領域を確保することもできます。ただし、この方法では各仮想デスクトップ上で個別に調整する必要があります。作成済みのマシンCatalogでVHDの初期サイズを調整することはできません。

VHDには、アンチウイルス定義ファイルを保存するのに十分なサイズを設定してください。通常、アンチウイルス定義ファイルのサイズは小さくありません。

HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\personal vDisk\Configで、以下のレジストリキーを設定します。これ以外のレジストリキーは変更しないでください。MinimumVHDSIZEInMBを除き、すべての値はマスターイメージ上で設定します（MinimumVHDSIZEInMBはマシン単位で変更できます）。マスターイメージ上で設定された値は、次回イメージ更新時に適用されます。

#### ● MinimumVHDSIZEInMB

Personal vDiskのアプリケーション領域（Cドライブ）の最小サイズをMB単位で指定します。新しいサイズは、既存のサイズよりも大きなものにする必要がありますが、ディスクのサイズからPvDReservedSpaceMBの値を差し引いたものよりも小さくする必要があります。

この値を大きくすると、Personal vDisk上のプロファイル領域からCドライブに空き領域が割り当てられます。この設定は、Cドライブの現在の使用サイズよりも小さい値を指定したり、EnableDynamicResizeOfAppContainerに0を設定したりすると無視されます。

デフォルト値：2048

#### ● EnableDynamicResizeOfAppContainer

動的サイズ変更アルゴリズムを有効または無効にします。

- 1を設定すると、アプリケーション領域（Cドライブ上）の空き容量が10%未満になったときにその領域のサイズが自動的に調整されます。1または0を設定できます。変更後、仮想マシンの再起動が必要です。
- 0を設定すると、XenDesktop 7.x以前で使用されていた方法でVHDのサイズが決定されます。

デフォルト値：1

#### ● EnableUserProfileRedirection

ユーザープロファイルのPersonal vDiskへのリダイレクトを有効または無効にします。

- 1を設定すると、ユーザーのプロファイルがPersonal vDiskドライブ（デフォルトでPドライブ）にリダイレクトされます。通常、各プロファイルは、P:\Usersフォルダーの、標準Windowsプロファイルに対応するサブフォルダーにリダイレクトされます。これにより、ユーザーのデスクトップのリセットが必要になった場合でもプロファイルが保持されます。
- 0を設定すると、Personal vDisk上のPvDReservedSpaceMBのサイズを差し引いた全領域がCドライブ（Personal vDiskのアプリケーション領域）に割り当てられ、Personal vDiskドライブ（Pドライブ）はWindowsエクスプローラーに表示されなくなります。Citrix Profile managementやほかの移動プロファイル管理ツールを使用する場合は、この値に0を設定してプロファイルのリダイレクトを無効にすることをお勧めします。

この設定により、プロファイルはPersonal vDiskにリダイレクトされずにC:\Users内に保持されます。このため、Profile managementなどの移動プロファイル管理ツールでプロファイルを管理できるようになります。

この設定により、Pドライブの領域がすべてアプリケーションに割り当てられます。

移動プロファイル管理ツールを使用する環境でのみ0を設定してください。移動プロファイル管理ツールを使用しない環境でこのレジストリ設定を使用すると、Personal vDiskをリセットしたときにプロファイルが削除されてしまいます。

この設定は、イメージの更新時には変更しないでください。イメージ更新時に設定値を1から0に変更すると既存のプロファイルが移動されないままPersonal vDiskの全領域がCドライブに割り当てられ、Personal vDiskが非表示になってしまいます。

この値は、マシンカタログを展開する前に設定してください。マシンカタログを展開した後で値を変更することはできません。

重要：XenDesktop 7.1以降では、イメージの更新時にこの値の変更が適用されなくなっています。このレジストリキーは、プロファイルの元になるカタログを最初に作成するときに設定してください。この設定を後で変更することはできません。

デフォルト値：1

#### ● PercentOfPvDForApps

Personal vDiskのアプリケーション領域（Cドライブ）とプロファイル領域との割合を指定します。この値は新しい仮想マシンの作成時に適用され、またEnableDynamicResizeOfAppContainerが0の場合はイメージの更新時にも適用されます。

PercentOfPvDForApps設定に対する変更を適用するには、EnableDynamicResizeOfAppContainerを0に設定する必要があります。EnableDynamicResizeOfAppContainerは、デフォルトで1（有効）に設定されています。このため、アプリケーション領域（AppContainer。表示上はCドライブ）は空き領域が10%未満になった時点で動的に拡張されます。

PercentOfPvDForAppsの値を増やしても、AppContainerの拡張が許可される最大領域が増えるだけです。設定した値がすぐに適用されるわけではありません。また、この割り当て比率の設定はマスターイメージ内で構成する必要があります。これにより、次回イメージ更新時に設定が反映されます。

EnableDynamicResizeOfAppContainerを1に設定したままマシンカタログを生成した場合は、マスターイメージ内でこの値を0に変更して、適切な割り当て比率を構成してください。構成する割り当て比率は、Cドライブの現在の割り当てサイズよりも大きな値である必要があります。

この値を0に設定すると、管理者が割り当て比率を完全に制御できます。つまり、ユーザーが消費する領域サイズにかかわらずCドライブのサイズを完全に制御でき、動的なサイズ調整は行われません。

デフォルト値：50% (2つの領域に同じサイズが割り当てられます)

- **PvDReservedSpaceMB**

Personal vDiskログおよびほかのデータ用に予約されるPersonal vDisk上の領域をMB単位で指定します。

XenApp 6.5 (またはそれ以前のバージョン) が動作する環境でアプリケーションのストリーム配信機能を使用する場合は、Rade Cacheのサイズに応じてこの値を増やしてください。

デフォルト値：512

- **PvDResetUserGroup**

Citrix XenDesktop 5.6にのみ適用され、Personal vDiskのリセットを許可するユーザーのグループを指定します。これ以降のバージョンのXenDesktopでは、管理権限の委任機能を使用します。

その他の設定：

- **Windows Updateサービス**：マスターイメージでは、Windows Updateのオプションとして [更新プログラムを確認しない] を選択し、Windows Updateサービスを [無効] に設定します。Windows Updateサービスを実行する必要がある場合でも、 [更新プログラムを確認しない] を選択することで更新プログラムがマシン上にインストールされることを防ぐことができます。  
たとえば、WindowsストアからWindowsストアアプリをインストールする場合は、このサービスを実行しておく必要があります。
- **Windowsの更新プログラム**：Internet Explorerを含むWindowsの更新プログラムをマスターイメージに適用しておきます。
- **再起動が必要な更新プログラム**：Windowsの更新プログラムの中には、インストールを完了するために何回かの再起動が必要になるものがあります。Personal vDiskインベントリを収集する前に、マスターイメージを正しく再起動して、適用した更新プログラムが完全にインストールされたことを確認してください。
- **アプリケーションの更新プログラム**：マスターイメージ上のアプリケーションに必要な更新プログラムを適用しておくことで、ユーザーのvDiskに必要なディスク領域を節約できます。また、各ユーザーのvDisk上のアプリケーションを個別に更新する手間も省けます。

一部のアプリケーションでは、Personal vDiskによるユーザー環境で問題が発生する場合があります。このような問題を避けるには、管理者がそれらのアプリケーションを個々のマシン上ではなくマスターイメージ上にインストールする必要があります。さらに、Personal vDisk環境で正しく動作する場合でも、特定の種類のアプリケーションについてはマスターイメージ上にインストールすることをお勧めします。

マスターイメージ上へのインストールが必須のアプリケーション：

- エージェントおよびクライアントソフトウェア (System Center Configuration Managerエージェント、App-V Client、



Citrix Receiverなど)

- 早期起動ドライバーをインストールまたは変更するアプリケーション
- プリンターやスキャナーのソフトウェアやドライバーをインストールするアプリケーション
- Windowsネットワークスタックを変更するアプリケーション
- VMware ToolsやXenServer Toolsなどの仮想マシンツール

マスターイメージ上へのインストールが推奨されるアプリケーション

- 多くのユーザーに配信するアプリケーション。以下のアプリケーションは、更新機能を無効にしてから配信します。
  - ボリュームライセンスを使用する、Microsoft OfficeやMicrosoft SQL Serverなどのエンタープライズアプリケーション。
  - Adobe Reader、Firefox、Chromeなど、ユーザーに共通のアプリケーション。
- SQL Server、Visual Studio、アプリケーションフレームワーク (.NETなど) などのサイズの大きなアプリケーション。

Personal vDiskのマシンにユーザーがインストールするアプリケーションについて、以下の推奨事項および制限事項があります。ただし、管理者権限を持つユーザーに対しては、一部の項目を強制できない場合があります。

- ユーザーがマスターイメージからアプリケーションをアンインストールして、そのアプリケーションを自分のPersonal vDisk上にインストールすることは避けてください。
- 管理者がマスターイメージのイメージ上のアプリケーションを更新したりアンインストールしたりするときは、十分に注してください。管理者がマスターイメージ上にアプリケーションをインストールした後で、そのアプリケーションバージョン用のアドオンソフトウェア（プラグインソフトウェアなど）をユーザーがインストールしている場合があります。このような依存関係が存在する場合、そのイメージ上のアプリケーションを更新したりアンインストールしたりすると、ユーザーのアドオンソフトウェアが正しく動作しなくなることがあります。たとえば、マスターイメージ上にインストールされているMicrosoft Office 2010に対応するVisio 2010をユーザーが自分のPersonal vDisk上にインストールした場合、マスターイメージ上のOfficeを更新するとローカルのVisioが動作しなくなることがあります。
- ハードウェア依存のライセンスを使用するソフトウェア（ dongleを使用したり署名ベースのハードウェアを使用したりするもの）はサポートされません。

Provisioning ServicesとPersonal vDiskを併用する場合は、以下の考慮事項があります。

- Studioの [管理者] ノードで、Soap Serviceアカウントを追加してマシン管理者以上の役割を割り当てます。これにより、Provisioning Services (PVS) のvDiskを実稼働段階に昇格するときにPersonal vDiskデスクトップが準備中の状態になります。
- Personal vDiskを更新するには、Provisioning Serviceのバージョン機能を使用する必要があります。更新したバージョンが実稼働段階に昇格するときに、Soap ServiceによりPersonal vDiskデスクトップが準備中状態になります。
- Personal vDiskのサイズは、Provisioning Serviceの書き込みキャッシュディスクよりも常に小さくなくてはなりません。Personal vDiskがProvisioning Serviceの書き込みキャッシュより小さいと、Personal vDiskディスクが書き込みキャッシュとして使用されてしまう場合があります。
- デリバリーグループを作成した後では、[Personal vDisk Image Update Monitoring Tool](#)（イメージ更新監視ツール）または[サイズ変更とプール統計のスクリプト](#)（personal-vdisk-poolstats.ps1）を使用してPersonal vDiskを監視することができます。

書き込みキャッシュディスクのサイズを正しく設定してください。Personal vDiskがアクティブな場合、ユーザーによる多くの書き込み処理（変更内容）がPersonal vDisk上にリダイレクトされます。このため、Provisioning Servicesの書き込みキャッシュディスクのサイズを小さく設定できる場合があります。ただし、Personal vDiskがアクティブでないとき（イメージ更新時など）にProvisioning Services書き込みキャッシュディスクのサイズが足りなくなり、マシンがクラッシュすることがあります。

Provisioning Servicesのベストプラクティスに従ってProvisioning Services書き込みキャッシュディスクのサイズを設定し、さ

らにマスターイメージ上のテンプレートVHDの2倍のサイズを統合（マージ）処理用に追加することをお勧めします。マージ処理ですべての領域が使用されることはまれですが、可能性はあります。

Personal vDiskが有効なマシンのカタログをProvisioning Servicesで展開する場合は、以下の点に注意してください。

- Provisioning Servicesのドキュメントの手順に従います。
- Studioのホスト接続の設定を編集して、同時操作を制限することができます。方法については、後述の説明を参照してください。
- アプリケーションやほかのソフトウェアをインストールまたはアップデートしてProvisioning Services vDiskを再起動した後でそのvDiskを更新した場合は、Personal vDiskインベントリを実行して仮想マシンをシャットダウンしてください。その後で、新しいバージョンを実稼働モードに昇格させます。そのカタログ内のPersonal vDiskデスクトップが自動的に準備中の状態になります。準備中にならない場合は、Soap ServiceアカウントにControllerのマシン管理者またはそれ以上の権限が付与されていることを確認してください。

Provisioning Servicesのテストモード機能を使用すると、更新済みのマスターイメージを使用するマシンのテストカタログを作成できます。このテストカタログで実用性をテストしてから、それを実稼働用に昇格させることができます。

Personal vDiskが有効なマシンのカタログをMachine Creation Services (MCS) で展開する場合は、以下の点に注意してください。

- XenDesktopのドキュメントの手順に従います。
- マスターイメージ作成後にPersonal vDiskインベントリを実行し、仮想マシンの電源を切ります（仮想マシンの電源を切らないとPersonal vDiskが正しく機能しません）。次に、マスターイメージのスナップショットを作成します。
- マシンカタログの作成ウィザードで、Personal vDiskのサイズとドライブ文字を指定します。
- デリバリーグループを作成した後では、[Personal vDisk Image Update Monitoring Tool](#)（イメージ更新監視ツール）または[サイズ変更とプール統計のスクリプト](#)（personal-vdisk-poolstats.ps1）を使用してPersonal vDiskを監視することができます。
- Studioのホスト接続の設定を編集して、同時操作を制限することができます。方法については、後述の説明を参照してください。
- マスターイメージを更新する場合は、マスターイメージ上のアプリケーションやほかのソフトウェアをアップデートした後でPersonal vDiskインベントリを実行し、仮想マシンの電源を切ります。次に、マスターイメージのスナップショットを作成します。
- Personal vDisk Image Update Monitoring Toolまたはpersonal-vdisk-poolstats.ps1スクリプトを使用して、更新したマスターイメージが展開される各仮想マシン上に十分な領域があることを確認します。
- マシンカタログを更新すると、各Personal vDiskデスクトップが準備中の状態になり、マスターイメージの更新内容が適用されます。各デスクトップは、マシン更新時に指定したロールアウト方法に基づいて更新されます。
- Personal vDisk Image Update Monitoring Toolまたはpersonal-vdisk-poolstats.ps1スクリプトを使用して、「準備中」状態のPersonal vDiskを監視します。

vDiskからファイルやフォルダーを除外するには、以下の規則ファイルを使用します。この方法は、展開済みのPersonal vDiskで使用できます。この規則ファイルの名前はcustom\_\*\_rules.template.txtで、configフォルダーに格納されています。これらのファイルの使用方法については、各規則ファイルのコメントを参照してください。

Personal vDiskを有効にしてマスターイメージを更新したら、ディスクのインベントリを更新し（この操作は「インベントリの実行」と呼ばれます）、新しいスナップショットを作成することが重要です。

マスターイメージを管理するのは（ユーザーではなく）管理者であるため、管理者がアプリケーションをインストールしたと

きにその管理者のプロファイルにバイナリファイルが配置されると、共有された仮想デスクトップ（プールされたマシンカタログおよびプールされたPersonal vDiskマシンカタログのデスクトップも含む）のユーザーがそのアプリケーションを使用できなくなります。このようなアプリケーションは、ユーザーが自分でインストールする必要があります。

以下の各手順を実行した後で、イメージのスナップショットを作成することをお勧めします。

1. マスターイメージを更新します。つまり、オペレーティングシステムの更新プログラムや必要なアプリケーションをインストールして、マシンのシステム構成を実行します。

Personal vDiskを使用するWindows XPベースのマスターイメージの場合は、ソフトウェアインストールの確認メッセージや未署名のドライバーの使用に対するメッセージなど、何らかのダイアログボックスが開いていないことを確認します。この環境のマスターイメージでダイアログボックスが開いていると、VDAをDelivery Controllerに登録できません。未署名のドライバーに対するメッセージは、コントロールパネルで無効にすることができます。たとえば、[システム]、[ハードウェア]、[ドライバの署名]の順に選択し、警告メッセージを無視するオプションを選択します。

2. マシンをシャットダウンします。Windows 7マシンの場合は、Citrix Personal vDiskがシャットダウンをブロックするときに[キャンセル]をクリックします。
3. [Citrix Personal vDisk] ダイアログボックスの[インベントリの更新]をクリックします。この処理が完了するまで数分間かかることがあります。  
重要：この処理の後のシャットダウンを中断すると、（軽微なイメージ更新であっても）Personal vDiskのインベントリがマスターイメージと一致しくなくなります。これによりPersonal vDiskが機能しくなくなります。シャットダウンを中断した場合は、マシンを再起動してから再度シャットダウンし、メッセージが表示されたら[インベントリの更新]をもう一度クリックします。
4. インベントリ操作によりマシンがシャットダウンしたら、マスターイメージのスナップショットを作成します。

インベントリをネットワーク共有上にエクスポートして、それをマスターイメージ上にインポートできます。詳しくは、「[Personal vDiskインベントリのエクスポートとインポート](#)」を参照してください。

デスクトップやアプリケーションを提供するマシンの電源状態は、Citrix Broker Serviceにより制御されます。このBroker Serviceは、Delivery Controllerを介していくつかのハイパーバイザーを制御することもできます。Broker Serviceの電源操作機能により、Controllerとハイパーバイザー間の相互操作が制御されます。ハイパーバイザーに過剰な負荷がかかることを防ぐため、マシンの電源状態に対する変更操作に優先度が割り当てられ、これによりハイパーバイザーの同時操作が制御されます。これを設定するには、次の手順に従います。これらの値を変更するには、Studioの[ホスト]ノードで[接続の編集]ダイアログボックスを開き、[詳細設定]ページを使用します。

ホスト接続の同時操作を制御するには、次の手順に従います。

1. Studioのナビゲーションペインで[構成]、[ホスト]の順に選択します。
2. 接続を選択し、[操作]ペインの[接続の編集]を選択します。
3. 必要に応じて、以下の値を変更します。
  - **同時操作（すべての種類）** - 同時に実行可能な電源操作の上限値を指定します。この値は、絶対値およびハイパーバイザーへの接続に対するパーセンテージで指定できます。2つの設定値のうちより小さい値が適用されます。  
デフォルト値：絶対値100、20%
  - **Personal vDiskストレージインベントリの同時操作** - 同時に実行可能なPersonal vDiskの電源操作の上限値を指定します。この値は、絶対値および接続に対するパーセンテージで指定できます。2つの設定値のうちより小さい値が適用されます。  
デフォルト値：絶対値50、25%

絶対値を計算するには、エンドユーザーのストレージでサポートされる合計IOPS（1秒あたりの読み取り/書き込み回数）を使用します。さらに、各仮想マシンのIOPS（IOPS/VM）を350として、ストレージで同時にアクティブにできる

仮想マシン数を計算します。合計IOPS値をIOPS/VM値で除算するとこの値が算出されます。

たとえば、エンドユーザーのストレージのIOPSが14000の場合、同時にアクティブにできる仮想マシン数は40 ( $14000 \div 350 = 40$ ) になります。

- **1分あたりの最大新規操作** - ハイパーバイザーに送信可能な新規電源操作の1分あたりの上限値を指定します。この値は、設定値で指定します。  
デフォルト値：10

これらの設定について最適な値を確認するには、以下の操作を行います。

1. デフォルトの値を使用して、テストカタログの単一イメージ更新にかかる合計応答時間を計測します。つまり、イメージ更新の開始時刻 (T1) からカタログ内の最後のマシンでのVDAのControllerへの登録時刻 (T2) までの時間を計測します (合計応答時間 = T2 - T1)。
2. イメージ更新時のハイパーバイザーストレージのIOPS (1秒あたりの読み取り/書き込み回数) を計測します。この値を最適化の基準値として使用します。通常はデフォルトの設定値を使用しますが、IOPSの限界まで達する場合はより小さい値を設定します。
3. 以下の手順に従って [Personal vDiskストレージインベントリの同時操作] の値を変更します (ほかのすべての設定値はそのまま保持します)。
  1. 値を10ずつ増やし、そのたびに合計応答時間を計測します。合計応答時間が低下または一定化するまでこれを繰り返します。
  2. 値を10ずつ増やしても合計応答時間が改善されない場合は、値を10ずつ減らし、そのたびに合計応答時間を計測します。合計応答時間が一定化し、改善されなくなるまでこれを繰り返します。これにより、最適なPersonal vDisk電源操作値を求めます。
4. 最適なPersonal vDisk電源操作値を確認したら、[同時操作 (すべての種類)] および [1分あたりの最大新規操作] の設定値を1つずつ調整します。これらの設定でも、上記の (値を10ずつ増減させる) 方法で値を変更して効果を確認します。

System Center Configuration Manager (Configuration Manager) 2012を使用する場合は特別な構成が不要で、ほかのマスターイメージアプリケーションと同じ方法でインストールできます。以下の説明は、System Center Configuration Manager 2007にのみ適用されます。Configuration Manager 2007より前のバージョンはサポートされません。

Personal vDisk環境でConfiguration Manager 2007エージェントソフトウェアを使用するには、以下の操作を行います。

1. マスターイメージにクライアントエージェントをインストールします。
  1. マスターイメージにConfiguration Managerクライアントをインストールします。
  2. ccmexec service (SMS Agent) を停止して、さらに無効に設定します。
  3. ローカルコンピューターの証明書ストアから、SMSまたはクライアント証明書を削除します。これを行うには、以下の手順に従います。
    - 混在モード：証明書 (ローカルコンピューター) \SMS\証明書
    - ネイティブモード
      - 証明書 (ローカルコンピューター) \個人\証明書
      - 証明機関 (通常は内部の公開キー基盤) により発行されたクライアント証明書を削除します。
  4. C:\Windows\smscfg.iniを削除するか、名前を変更します。
2. クライアント固有の情報を削除します。
  1. C:\Windows\System32\CCM\Loggsのログファイルを削除または移動します (オプション)。
  2. Virtual Delivery Agentがインストールされていない場合はインストールして、Personal vDiskのインベントリを実行します。
  3. マスターイメージをシャットダウンしてスナップショットを作成し、このスナップショットを使用してマシンカタログを作成します。

3. Personal vDiskを検証して、サービスを起動します。各Personal vDiskデスクトップの初回起動時に、以下の手順を1回実行します。ドメインのグループポリシーオブジェクトを使用してこれを実行することもできます。

- Personal vDiskがアクティブであることを確認します。これを行うには、レジストリキー HKEY\_LOCAL\_MACHINE\Software\Citrix\personal vDisk\config\virtualが存在することを確認します。
- ccmexec service (SMSエージェント) を [自動] にして、このサービスを起動します。Configuration ManagerクライアントがConfiguration Managerサーバーと通信して、新しい固有の証明書およびGUIDが取得されます。

# ツール

Jan 04, 2017

以下のツールおよびユーティリティを使ってPersonal vDiskの機能を構成、管理、および監視できます。

Personal vDiskに付属の規則を使用して、Personal vDiskイメージの以下のデフォルトの動作を変更できます。

- Personal vDisk上のファイルの表示/非表示
- ファイルに対する変更内容のマージ方法
- ファイルの書き込みの許可/禁止

カスタムの規則ファイルおよびコピーオンライトについて詳しくは、各ファイル内に記述されているコメントを参照してください。これらのファイルは、Personal vDiskをインストールしたマシン(C:\ProgramData\Citrix\personal vDisk\Config)にあります。「custom\_\*」という名前のファイルには、規則の説明と規則を有効にする方法が記述されています。

Personal vDiskのサイズを監視したり管理したりするための2つのスクリプトが用意されています。これらのスクリプトは、XenDesktopインストールメディアのSupport\Tools\Scriptsフォルダーに収録されています。また、Support\Tools\Scripts\PvdToolフォルダーに収録されているPersonal vDisk Image Update Monitoring Tool (イメージ更新監視ツール)を使用することもできます。詳しくは、<http://blogs.citrix.com/2014/06/02/introducing-the-pvd-image-update-monitoring-tool/>を参照してください。

resize-personalvdisk-pool.ps1スクリプトでは、カタログ内のすべてのデスクトップのPersonal vDiskのサイズを増やすことができます。このスクリプトを使用するには、Studioが動作するマシン上に、使用するハイパーバイザーに対応する以下のスナップインまたはモジュールをインストールする必要があります。

- XenServer : XenServerPSSnapin
- vCenter : vSphere PowerCLI
- System Center Virtual Machine Manager : VMMコンソール

personal-vdisk-poolstats.ps1スクリプトでは、複数のPersonal vDiskに対してアプリケーション領域とユーザープロファイル領域の容量を確認したり、イメージの更新状態を確認したりできます。イメージを更新する前にこのスクリプトを実行すると、更新の失敗の原因となる空き容量不足のデスクトップを特定することができます。このスクリプトを使用するには、Personal vDiskデスクトップのファイアウォールで、Windows Management Instrumentationからの受信規則 (WMI受信) が有効になっている必要があります。この規則は、マスターイメージまたはGPOを使用して有効にできます。

イメージの更新に失敗すると、[Update] 列にその原因が表示されます。

問題のあるアプリケーションのインストールなどが原因でユーザーのデスクトップが破損した場合は、管理者がPersonal vDiskのアプリケーション領域を工場出荷時のデフォルト (つまり空の状態) にリセットできます。この領域をリセットしても、ユーザープロファイルには影響しません。

Personal vDiskのアプリケーション領域をリセットするには、以下のいずれかを行います。

- ユーザーのデスクトップに管理者としてログオンします。コマンドラインで\Program Files\Citrix\Personal vDisk\bin\CtxPvD.exe -s Resetを実行します。
- Citrix Directorでユーザーのデスクトップを選択します。[Personal vDiskをリセット] をクリックして [OK] をクリックします。

イメージの更新プロセスは、Personal vDiskデスクトップに新しいイメージをロールアウトする操作に組み込まれています。これにより、新しい基本イメージで既存のPersonal vDiskを使用できるようになります。Machine Creations Services (MCS) を使用する展開環境では、アクティブな仮想マシンからインベントリをネットワーク共有上にエクスポートして、それをマスターイメージ上にインポートできます。マスターイメージでは、このインベントリ情報に基づいて差分が計算されます。インベントリのエクスポート/インポート機能を使用することは必須ではありませんが、これによりイメージ更新プロセスの全体的なパフォーマンスが向上します。

インベントリのエクスポート/インポート機能を使用するには、管理者権限が必要です。必要に応じてnet useコマンドを実行して、エクスポート/インポート機能で使用するネットワーク共有をマップします。このとき、エクスポートまたはインポートで使用するすべてのファイルにアクセスできるアカウントが必要です。

- インベントリをエクスポートするには、Personal vDisk (Version 7.6以降) が有効なVDAのマシン上で、管理者として次のエクスポートコマンドを実行します。

```
Ctxpvdsv.exe exportinventory "<path-to-export-location>"
```

現在のインベントリの場所が検出され、で指定した場所のExportedPvdInventoryフォルダーにインベントリがエクスポートされます。このコマンドにより、以下のようなメッセージが出力されます。

```
C:\Program Files\Citrix\personal vDisk\bin> .\CtxPvDsv.exe exportinventory \\share location\ExportedInventory Current inventory source location C:\Citrix\PVD\Settings\Inventory\VER-LAS ... Exporting
```

- エクスポートしたインベントリをマスターイメージにインポートするには、管理者としてインポートコマンドを実行します。

インベントリをインポートするには

マスターイメージ上で、管理者として次のインポートコマンドを実行します。

```
Ctxpvdsv.exe importinventory "<path-to-exported-inventory>"
```

<path to exported inventory>には、エクスポート済みのインベントファイルのフルパス (通常<network location>\ExportedPvdInventory) を指定します。

exportinventoryオプションでエクスポートされたインベントリが取得され、それがマスターイメージ上のインベントリストアにインポートされます。このコマンドにより、以下のようなメッセージが出力されます。

```
C:\Program Files\Citrix\personal vDisk\bin> .\CtxPvDsv.exe importinventory \\share location\ExportedInventory\ExportedPvdInventory Importing inventory \\share location\ExportedInventory\ExportedPvdInventory
```

インベントリのエクスポート先には、以下のファイルが出力されます。これらのファイルは、マスターイメージ上のインベントリストアに同じ名前でもインポートされます。

- Components.DAT
- files\_rules
- folders\_rules
- regkey\_rules
- RINGTHREE.DAT
- S-1-5-18.DAT
- SAM.DAT
- SECURITY.DAT
- SNAPSHOT.DAT

- SOFTWARE.DAT
- SYSTEM.CurrentControlSet.DAT
- VDCATALOG.DAT
- vDiskJournalData

重要：これらのスクリプトを実行しても、Personal vDiskのストレージの場所を変更することはできません。これを行うには、ほかの方法を使用してください。

インストールメディアのSupport\Tools\Scriptsフォルダーに収録されている2つのPowerShellスクリプトを使用して、Personal vDisksをバックアップして復元することができます。バックアップと復元用のスクリプトを使用して、既存のPersonal vDiskとユーザーの関連付けを別のカタログに移行できます。この方法は、Personal vDiskストレージを変更するときにも使用できます。バックアップスクリプトを実行すると、既存のカタログのメタデータを含むXMLファイルが作成されます。このメタデータには、Personal vDiskの現在のストレージの場所と関連付けられているユーザーの情報が含まれます。このXMLファイルは復元スクリプトで使用すると、Personal vDiskが新しいカタログに関連付けられ、ユーザーが割り当てられます。

- migration-backup.ps1は、各ユーザーとマシンカタログ内のPersonal vDisk間のマッピング情報をXMLファイルとして保存します。
- migration-restore.ps1は、そのXMLファイルを使ってマシンカタログ内でユーザーのデスクトップを再作成します。

バックアップや復元を実行する前に、以下の点に注意してください。

- これらのスクリプトではハイパーバイザーAPIを使用するため、スクリプトを実行するControllerにハイパーバイザーのPowerShellスナップインをインストールしておく必要があります。
- 対象のマシンカタログのControllerにアクセスできる場所でスクリプトを実行する必要があります。
- これらのスクリプトがサポートするハイパーバイザープラットフォームは、Citrix XenServer、Microsoft Hyper-V、およびVMware ESXです。

### マシンカタログのバックアップ

マシンカタログを変更するたびにバックアップを作成してください。カタログ内のマシンが使用中の場合でもバックアップを実行できます。

Personal vDiskを持つマシンカタログをバックアップするには、migration-backup.ps1を使用します。このスクリプトの実行時に、マシンカタログ名とハイパーバイザーの接続情報を入力する必要があります。すると、マシンカタログ内の各マシンについてPersonal vDiskストレージと割り当てられたユーザーの間のマッピング情報が保存されます。この情報はXMLファイルに格納されます。このXMLファイルの構造は以下のとおりです。

- PvdMigration.hypervisor.Typeは、ハイパーバイザーの種類（VMware ESX、Citrix XenServer、またはMicrosoft Hyper-V）です。
- PvdMigration.PVDは、Personal vDiskの格納場所と、そのPersonal vDiskに割り当てられているユーザーの情報です。
- PvdMigration.PVD.DiskIdは、バックアップしたハイパーバイザー上のvDiskの一意なIDです。
- PvdMigration.PVD.DiskNameは、VHDまたはVMDKファイルの名前です。
- PvdMigration.PVD.SRNameは、バックアップを実行したときのストレージプロバイダーの名前です。
- PvdMigration.PVD.SRIDは、バックアップしたハイパーバイザー上のストレージプロバイダーの一意なIDです。
- PvdMigration.PVD.UserNameは、このvDiskに割り当てられているユーザーの名前です。
- PvdMigration.PVD.UserSidは、このvDiskに割り当てられているユーザーのSIDです。
- PvdMigration.PVD.Stateは、このvDiskの状態を示し、「backed up」または「processed」のいずれかです。初回のバックアップの後は「backed up」になります。このXMLファイルを使ってバックアップを復元すると、状態が「processed」に変更されます。

### マシンカタログの復元

復元を実行する前に、次の点に注意してください。

- バックアップしたマシンカタログと同じマスターイメージを共有するマシンカタログだけを復元できます。
- バックアップしたマシンカタログのマスターイメージのインベントリを更新して、新しいマスターイメージを作成する必要があります。

Personal vDiskを持つマシンカタログをバックアップから復元するには、migration-restore.ps1を使用します。このスクリプトの実行時に、以下の情報を入力する必要があります。

- バックアップ時に生成されたXMLファイル。
- 復元するマシンカタログの名前。
- 未アタッチのPersonal vDiskの格納場所の名前。この情報は、バックアップ時のXMLファイル内に記述されています。
- ハイパーバイザーの接続情報

migration-restore.ps1スクリプトはマシンカタログ内の未割り当てマシンを検索し、それにユーザーを割り当てます。また、ユーザーのPersonal vDiskをマシンにアタッチします。

### シナリオ例1 — 新しいマシン名を使用したマシンカタログおよびそのPersonal vDiskの復元

このシナリオでは、マシンカタログ全体とそのマシンにアタッチされたPersonal vDiskが復元されます。各マシンには新しい名前が指定されます。以下の手順は、ハイパーバイザーやストレージホストで障害が生じた場合、または新しいインフラストラクチャにユーザーを移行する場合に使用します。

1. migration-backup.ps1を実行して、ユーザーとPersonal vDiskとのマッピングをXMLファイルにバックアップします。
2. 何らかのバックアップソリューションを使用して、元のマシンカタログから任意のディスク上にPersonal vDiskを移動またはキャプチャします。
  - VMware ESXまたはMicrosoft Hyper-Vの場合：Personal vDiskは、Controllerにより指定されたストレージ上の、vDiskのアタッチ先マシン名を示すフォルダー内にあります。
  - Citrix XenServerの場合：Personal vDiskは、Controllerにより指定されたストレージのルートにあります。各vDiskの名前はGUIDです。
3. 何らかのストレージバックアップソリューションを使用して、元のマシンカタログからPersonal vDiskを復元します。
  - ESXまたはHyper-Vの場合：新しいストレージリソースの新しいフォルダーにvDiskを配置します。または、新しいストレージリソース上で、バックアップ元と同じパスにvDiskを配置します。
  - XenServerの場合：新しいストレージリソース上のルートにvDiskを配置します。
4. 障害が発生したマシンカタログの作成元のマスターイメージから、Provisioning Services vDiskまたはMachine Creation Servicesスナップショットを作成します。
5. vDiskまたはスナップショット上の [スタート] メニューから [インベントリの更新] を実行します。
6. Studioでマシンカタログを再作成します。このとき、障害が発生した（オリジナルの）マシンカタログと異なる規則で名前を指定します。これにより、新しいPersonal vDiskを持つ新しいマシンカタログが作成され、サイトデータベースに認識されます。
7. 再作成されたマシンカタログが正しいデリバリーグループに割り当てられていることを確認します。
8. デリバリーグループがメンテナンスモードで、そのグループの各マシンがシャットダウン状態であることを確認します。
9. バックアップスクリプトにより生成されたXMLファイルを編集します。
  - ESXまたはHyper-Vの場合：手順3で新しいストレージリソース上の新しいフォルダーにvDiskを復元した場合、ファイル内のすべてのセクションについて、のフォルダー名を新しいフォルダー名に置き換えます。バックアップ元と同じパスにvDiskを復元した場合は、この手順をスキップします。
  - XenServerの場合：この手順をスキップします。
10. Controller上でmigration-restore.ps1を実行し、XMLファイルの名前およびバックアップされたvDiskの格納場所を指定します。

### シナリオ例2 — 既存のマシン名を再使用したマシンカタログおよびそのPersonal vDiskの復元

このシナリオでは、マシンカタログ全体とそのマシンにアタッチされたPersonal vDiskが復元されます。各マシンには、元の名前が再使用されます。以下の手順は、ハイパーバイザーやストレージホストで障害が生じた場合に使用します。

1. migration-backup.ps1を実行して、ユーザーとPersonal vDiskとのマッピングをXMLファイルにバックアップします。
2. 何らかのバックアップソリューションを使用して、元のマシンカタログから任意のディスク上にPersonal vDiskを移動またはキャプチャします。
  - ESXまたはHyper-Vの場合：Personal vDiskは、Controllerにより指定されたストレージ上の、vDiskの\_attach先マシン名を示すフォルダー内にあります。
  - XenServerの場合：Personal vDiskは、Controllerにより指定されたストレージのルートにあります。各vDiskの名前はGUIDです。
3. 何らかのストレージバックアップソリューションを使用して、元のマシンカタログからPersonal vDiskを復元します。
  - ESXまたはHyper-Vの場合：新しいストレージリソースの新しいフォルダーにvDiskを配置します。
  - XenServerの場合：新しいストレージリソース上のルートにvDiskを配置します。
4. 障害が発生したマシンカタログの作成元のマスターイメージから、Provisioning Services vDiskまたはMachine Creation Servicesスナップショットを作成します。
5. vDiskまたはスナップショット上の [スタート] メニューから [インベントリの更新] を実行します。
6. Studioでマシンカタログを再作成します。このとき、障害が発生した (オリジナルの) マシンカタログと同じ規則で名前を指定します。これにより、新しいPersonal vDiskを持つ新しいマシンの：
  - タログが作成され、サイトデータベースに認識されます。
7. 再作成されたマシンカタログが正しいデリバリーグループに割り当てられていることを確認します。
8. デスクトップグループがメンテナンスモードで、そのグループの各マシンがシャットダウン状態であることを確認します。
9. バックアップスクリプトにより生成されたXMLファイルを編集します。
  - ESXまたはHyper-Vの場合：ファイル内のすべてのセクションについて、のフォルダー名をvDiskの復元先フォルダー名に置き換えます。
  - XenServerの場合：この手順をスキップします。
10. Controller上で、編集したXMLファイルを指定してmigration-restore.ps1スクリプトを実行します。スクリプトにより、vDiskが移動せずにアタッチされます。
11. ユーザーデータが正しく復元されていることを確認します。

### シナリオ例3 — マシンカタログの一部のPersonal vDiskの復元

このシナリオでは、マシンカタログ内の一部のPersonal vDiskで障害が発生したため、そのvDiskだけを復元します。カタログ内の仮想マシンでは障害が発生していません。

1. migration-backup.ps1を実行して、ユーザーとPersonal vDiskとのマッピングをXMLファイルにバックアップします。
2. XMLファイルには、マシンカタログの各ユーザーに対応するセクションがあります。Personal vDiskを復元する必要がないユーザーについて、ファイルからそれらのユーザーおよび関連セクションを削除します。
3. ほかのシナリオで説明しているように、何らかのバックアップソリューションを使って元のマシンカタログからPersonal vDiskを復元します。
  - 新しいマシン名を使用する場合は、シナリオ例1を参照してください。
  - マシン名を保持する場合は、シナリオ例2を参照してください。
4. カタログに十分な未割り当てマシンがあることを確認します。必要な場合は、マシンを追加します。vDiskを復元する各ユーザーについて、新しいマシンが1つ必要です。
5. デスクトップグループがメンテナンスモードで、そのグループの各マシンがシャットダウン状態であることを確認します。
6. Controller上で、編集したXMLファイルを入力として指定してmigration-restore.ps1スクリプトを実行します。
7. ユーザーデータが正しく復元されていることを確認します。



# 表示、メッセージ、およびトラブルシューティング

Jan 04, 2017

Personal vDiskが有効なマシンをStudioのマシンカタログで選択すると、[PvD] タブにイメージ更新時の監視状態や推定完了時間、および進行状況が表示されます。イメージ更新時には、準備完了、準備中、待機中、失敗、および要更新のいずれかの状態が表示されます。

イメージの更新は、ディスクの空き容量不足やPersonal vDiskが見つからないなど、さまざまな要因で失敗することがあります。Studioでイメージ更新に失敗したことが示されると、トラブルシューティングに役立つエラーコードおよび説明が表示されます。Personal vDisk Image Update Monitoring Tool (イメージ更新監視ツール) またはpersonal-vdisk-poolstats.ps1スクリプトを使用すると、イメージの更新状況を監視して、問題が生じた場合はそのエラーコードを取得できます。

イメージ更新に失敗した場合は、以下のログファイルを参照してトラブルシューティングを行います。

- Personal vDiskサービスログ : C:\ProgramData\Citrix\personal vDisk\Logs\PvDsvc.log.txt
- Personal vDiskアクティブ化ログ : P:\PVDLOGS\PvDActivation.log.txt

最新の情報は、これらのログファイルの末尾に記録されます。

Personal vDisk 7.6およびそれ以降では、以下のエラーメッセージが生成されます。

- **内部エラーが発生しました。** 詳しくは、Personal vDiskのログを参照してください。エラーコード%d (%s)  
このメッセージは未分類のエラーに対して生成され、特定のエラーコードは提供されません。インベントリ作成またはPersonal vDisk更新時に予期されないエラーが発生すると、このメッセージが生成されます。
  - ログファイルを収集してCitrixのサポート担当者にお問い合わせください。
  - カタログ更新時にこのエラーが発生した場合は、カタログをロールバックして以前のゴールデンイメージの状態に戻してください。
- **規則ファイルに構文エラーがあります。** 詳しくは、ログを参照してください。  
エラーコードは、2です。規則ファイルに構文エラーが含まれています。Personal vDiskのログファイルに、規則ファイルの名前と構文エラーの行番号が記録されます。規則ファイルの構文エラーを修正して再試行してください。
- **前のバージョンのマスターイメージに対応しているPersonal vDisk上のインベントリが破損しているか、読み取ることができません。**  
エラーコードは、3です。最新のインベントリは、\ProgramData\CitrixPvD\Settings\Inventory\VER-LASTのUserData.V2.vhdに格納されます。正常なPersonal vDiskマシンから「VER-LAST」フォルダーをインポートして、マスターイメージの最終バージョンに相当するインベントリを復元してください。
- **前のバージョンのマスターイメージに対応しているPersonal vDisk上のインベントリは、より新しいバージョンです。**  
エラーコードは、4です。前回のマスターイメージと新しいマスターイメージでPersonal vDiskのバージョンが異なるとこの問題が発生します。マスターイメージに最新バージョンのPersonal vDiskをインストールしてからカタログ更新を再試行してください。
- **変更ジャーナルのオーバーフローが検出されました。**  
エラーコードは、5です。インベントリ作成時にマスターイメージに加えられた変更の数が多いと、USNジャーナルのオーバーフローが発生します。何回か再試行してもこのエラーが発生する場合は、procmonを使用して、インベントリ作成時サードパーティソフトウェアが大量のファイルを作成したり削除したりしていないかどうかを確認してください。
- **Personal vDiskで、ユーザーデータの格納用にシステムにアタッチされたディスクが見つかりません。**  
エラーコードは、6です。まず、ハイパーバイザーのコンソールでPersonal vDisk用のディスクが仮想マシンにアタッチ

(接続) されていることを確認してください。一般的に、このエラーはデータの漏洩を防ぐソフトウェアにより Personal vDisk用ディスクへのアクセスが阻止されると発生します。Personal vDisk用のディスクが仮想マシンにアタッチされている場合は、データ漏洩を防ぐソフトウェアにそのディスクに対する除外規則を追加してください。

- **インストール後にシステムが再起動されていません。変更内容を実装するために再起動してください。**  
エラーコードは、7です。デスクトップを再起動してから再試行してください。
- **インストールが破損しています。 Personal vDiskを再インストールしてください。**  
エラーコードは、8です。Personal vDiskを再インストールしてから再試行してください。
- **Personal vDiskのインベントリが最新の状態ではありません。 マスターイメージ内でインベントリを更新してから再試行してください。**  
エラーコードは、9です。デスクトップのシャットダウン前にマスターイメージ内でPersonal vDiskインベントリが更新されていないとこのエラーが発生します。マスターイメージを再起動して、[Personal vDiskの更新] オプションを使用してデスクトップをシャットダウンし、その後で新しいスナップショットを作成してください。このスナップショットを使用してカタログを更新します。
- **Personal vDiskの起動時に内部エラーが発生しました。 詳しくは、 Personal vDiskのログを参照してください。**  
エラーコードは、10です。このエラーは、内部エラーまたはPersonal vDiskの破損により、Personal vDiskドライバーで仮想化セッションの起動に失敗すると発生します。Controllerを使用してデスクトップを再起動してください。問題が解決しない場合は、ログファイルを収集してCitrixのサポート担当者にお問い合わせください。
- **ユーザーのパーソナル設定を格納するディスクの検索時にPersonal vDiskでタイムアウトが発生しました。**  
エラーコードは、11です。このエラーは、再起動後30秒以内にPersonal vDiskドライバーでPersonal vDisk用のディスクを検出できない場合に発生します。通常、サポートされない種類のSCSIコントローラーを使用したり、ストレージで遅延が発生したりすると検出に失敗します。マシンカタログ内のすべてのデスクトップでこのエラーが発生する場合は、テンプレート仮想マシンやマスター仮想マシンのSCSIコントローラーの種類をPersonal vDiskでサポートされるものに変更してください。このエラーがマシンカタログ内の一部のデスクトップのみで発生する場合は、多くのデスクトップの同時起動などで一時的にストレージに遅延が発生していることが考えられます。ホスト接続の設定で、最大電源操作数を制限することを検討してください。
- **システムが正しくシャットダウンされなかったためPersonal vDiskが無効になっています。 マシンを再起動してください。**  
エラーコードは、12です。このエラーは、Personal vDiskが有効なデスクトップで起動プロセスを完了できない場合に発生します。デスクトップを再起動してください。問題が解決しない場合は、ハイパーバイザーのコンソールでデスクトップの起動状況を監視して、デスクトップがクラッシュしているかどうかを確認します。デスクトップが起動中にクラッシュする場合は、バックアップ (バックアップがある場合) からPersonal vDiskを復元するか、Personal vDiskをリセットしてください。
- **Personal vDiskをマウントするためのドライブ文字を使用できません。**  
エラーコードは、13です。このエラーは、管理者により指定されたマウントポイントにPersonal vDisk用のディスクをマウントできない場合に発生します。指定されたドライブ文字がほかのハードウェアにより使用されていると、Personal vDisk用ディスクのマウントに失敗します。Personal vDisk用にほかのマウントポイントを指定してください。
- **Personal vDiskカーネルモードドライバーのインストールに失敗しました。**  
エラーコードは、14です。Personal vDiskをインストールした後の初回インベントリ更新時に、ドライバーがインストールされます。一部のアンチウイルス製品では、インストーラーコンテキスト外のドライバーのインストールがブロックされることがあります。初回インベントリ作成時にアンチウイルス製品のリアルタイムスキャン機能を一時的に無効にするか、例外規則を追加してください。

- システムボリュームのスナップショットを作成できません。Volume Shadow Copyサービスが有効になっていることを確認してください。  
エラーコードは、15です。このエラーは、Volume Shadow Copyサービスが無効になっていると発生します。Volume Shadow Copyサービスを有効にしてからインベントリ操作を再試行してください。
- 変更ジャーナルのアクティブ化に失敗しました。数分待ってから再試行してください。  
エラーコードは、16です。Personal vDiskでは、マスターイメージに対する変更を追跡するために「変更ジャーナル」が使用されます。インベントリ更新時に変更ジャーナルが無効になっていることが検出されると、有効化が試行され、有効化に失敗するとこのエラーが発生します。しばらくしてから再試行してください。
- システムボリュームに十分な空き領域がありません。  
エラーコードは、17です。デスクトップのCドライブにイメージの更新操作に必要な空き領域がない場合にこのエラーが発生します。システムボリュームを拡張するか、不要なファイルを削除して空き領域を確保してください。十分な空き領域を確保すると、次回再起動時にイメージ更新が開始されます。
- Personal vDiskストレージに十分な空き領域がありません。Personal vDiskストレージを拡張して空き領域を増やしてください。  
エラーコードは、18です。イメージの更新時にPersonal vDisk用のドライブに十分な空き領域がない場合にこのエラーが発生します。Personal vDiskストレージを拡張するか、不要なファイルを削除して空き領域を確保してください。十分な空き領域を確保すると、次回再起動時にイメージ更新が再開されます。
- Personal vDiskストレージがオーバーコミットされました。Personal vDiskストレージを拡張して空き領域を増やしてください。  
エラーコードは、19です。シックプロビジョニングされたUserData.V2.vhdを格納するための空き領域がPersonal vDisk用のドライブにない場合にこのエラーが発生します。Personal vDiskストレージを拡張するか、不要なファイルを削除して空き領域を確保してください。
- システムレジストリが破損しています。  
エラーコードは、20です。システムレジストリが破損または欠落しているか、読み取り不能になっています。Personal vDiskをリセットするか、作成済みのバックアップから復元してください。
- Personal vDiskのリセット時に内部エラーが発生しました。詳しくは、Personal vDiskのログを参照してください。  
エラーコードは、21です。このメッセージは、Personal vDiskのリセット時に発生したすべてのエラーに対して生成されます。ログファイルを収集してCitrixのサポート担当者にお問い合わせください。
- Personal vDiskのリセットに失敗しました。Personal vDiskストレージに十分な空き領域がありません。  
エラーコードは、22です。リセット時にPersonal vDisk用のドライブに十分な空き領域がない場合にこのエラーが発生します。Personal vDiskストレージを拡張するか、不要なファイルを削除して空き領域を確保してください。

Version 7.6より前のPersonal vDisk 7.xでは、以下のエラーメッセージが生成されます。

- スタートアップに失敗しました。Personal vDiskは、ユーザーの個人設定用のストレージディスクを見つけることができませんでした。  
このエラーメッセージは、Personal vDiskソフトウェアでPersonal vDisk（デフォルトでPドライブ）が見つからない場合、または管理者がカタログ作成時に指定したマウントポイントにそのディスクをマウントできない場合に表示されます。
  - この問題が発生した場合は、Personal vDiskサービスログで「PvD 1 status --> 18:183」を検索します。
  - Version 5.6.12よりも古いバージョンのPersonal vDiskを使用している場合は、最新バージョンにアップグレードすることでこの問題を解決できます。
  - Version 5.6.12またはそれ以降のバージョンを使用している場合は、ディスクの管理ツール（diskmgmt.msc）を使用して

Pドライブが不明なボリュームとして存在することを確認してください。Pドライブが存在する場合は、そのボリューム上でchkdskを実行します。ボリュームが破損していることが検出された場合は、chkdskを使用して修復してください。

- スタートアップに失敗しました。Citrix Personal vDiskを起動できませんでした。詳細は .... 状態コード：7、エラーコード0x70

「状態コード7」はPersonal vDisk更新時のエラーを示します。次のいずれかのエラーコードが表示されます。

| エラーコード     | 説明                                                                                                |
|------------|---------------------------------------------------------------------------------------------------|
| 0x20000001 | 差分パッケージの保存に失敗しました。VHDの空き領域不足が考えられます。                                                              |
| 0x20000004 | Personal vDiskの更新に必要な特権の取得に失敗しました。                                                                |
| 0x20000006 | Personal vDiskイメージまたはPersonal vDiskインベントリからのハイブのロードに失敗しました。Personal vDiskイメージまたはインベントリの破損が考えられます。 |
| 0x20000007 | ファイルシステムインベントリのロードに失敗しました。Personal vDiskイメージまたはインベントリの破損が考えられます。                                  |
| 0x20000009 | ファイルシステムインベントリを含んでいるファイルを開くことができません。Personal vDiskイメージまたはインベントリの破損が考えられます。                        |
| 0x2000000B | 差分パッケージの保存に失敗しました。VHDの空き領域不足が考えられます。                                                              |
| 0x20000010 | 差分パッケージのロードに失敗しました。                                                                               |
| 0x20000011 | 規則ファイルがありません。                                                                                     |
| 0x20000021 | Personal vDiskインベントリが破損しています。                                                                     |
| 0x20000027 | カタログ「MojoControl.dat」が破損しています。                                                                    |
| 0x2000002B | Personal vDiskインベントリが破損または欠落しています。                                                                |
| 0x2000002F | 更新時に、ユーザーによりインストールされたMOFの登録に失敗しました。Version 5.6.12にアップグレードすることで解決できます。                             |
| 0x20000032 | PvDactivation.log.txtで、最後のWin32エラーコードエントリを確認してください。                                               |
| 0x20       | イメージ更新用のアプリケーションコンテナのマウントに失敗しました。Version 5.6.12にアップグレードすることで解決できます。                               |
| 0x70       | ディスク上に十分な領域がありません。                                                                                |

- スタートアップに失敗しました。Citrix Personal vDiskを起動できませんでした。（または「Personal vDiskで内部エラーが発生しました。」） 詳細は .... 状態コード：20、エラーコード：0x20000028  
このメッセージは、Personal vDiskが見つかったにもかかわらずPersonal vDiskセッションを作成できなかったことを示します。

ログを収集して、SysVol-IvmSupervisor.logにセッションの作成の失敗が記録されていないかどうかを確認してください。

1. 「IvmNativeSessionCreate: failed to create native session, status」というエントリを検索します。
  2. が0xc00002cfの場合は、カタログに新しいバージョンのマスターイメージを追加することで解決できます。この状態コードは、インベントリ更新後の変更数が多いためにUSNジャーナルのオーバーフローが発生したことを示します。
  3. 問題が発生した仮想デスクトップを再起動します。問題が解決されない場合は、Citrixのテクニカルサポートに問い合わせてください。
- スタートアップに失敗しました。安全ではないシステムシャットダウンが検出されたため、Citrix Personal vDiskが非アクティブ化されています。もう一度実行するには、[再試行] をクリックします。問題が解決しない場合は、システム管理者に連絡してください。

このメッセージは、プールされた仮想マシンを、Personal vDiskを有効にしたまま起動できないことを示します。まず、起動に失敗した原因を調べます。考えられる原因として、以下の理由によるブルースクリーンエラーが挙げられます。

- 互換性のないアンチウイルスソフトウェア（古いバージョンのTrend Microなど）がマスターイメージ上にインストールされている。
- Personal vDiskと互換性のないソフトウェアがユーザーによりインストールされている。これが原因となることはまれですが、マシンカタログに新しいマシンを追加して、そのマシンが正しく再起動するかどうかを確認してください。
- Personal vDiskイメージが破損している（この問題はVersion 5.6.5で確認されています）。

プールされた仮想マシンでブルースクリーンエラーが発生するかどうか、または不完全な状態で起動するかどうかを確認するには、以下の操作を行います。

- ハイパーバイザーのコンソールを使用して、仮想マシンにログオンします。
- [再試行] をクリックして、仮想マシンがシャットダウンするまで待ちます。
- Studioを使用して、仮想マシンを起動します。
- ハイパーバイザーのコンソールを使用して、仮想マシンの起動時に問題が発生するかどうかを確認します。

また、以下の解決方法があります。

- 仮想マシンからメモリダンプを収集して、それをCitrix社のテクニカルサポート部門に送ります。
- 次の手順で、Personal vDiskのイベントログにエラーが記録されているかどうかを確認します。
  1. DiskMgmt.mscを起動して、[操作] メニューの [VHDの接続] を選択して、PドライブのルートにあるUserData.V2.vhdをマウントします。
  2. Eventvwr.mscを起動します。
  3. [操作] メニューの [保存されたログを開く] を選択して、UserData.V2.vhdのシステムイベントログ (Windows\System32\winevt\logs\system.evtx) を開きます。
  4. [操作] メニューの [保存されたログを開く] を選択して、UserData.V2.vhdのアプリケーションイベントログ (Windows\System32\winevt\logs\application.evtx) を開きます。

- Personal vDiskを開始できません。インベントリが更新されていないため、Personal vDiskを開始できませんでした。マスターイメージのインベントリを更新してから再試行してください。状態コード：15、エラーコード0x0  
このメッセージは、管理者がPersonal vDiskカタログを作成するときに不適切なスナップショットを選択した（つまりスナップショット作成時にマスターイメージが [Update Personal vDisk] でシャットダウンされなかった）ことを示します。

Personal vDiskが無効な場合は、Windowsイベントビューアーで以下のイベントを確認できます。Personal vDisk関連のイベントは [アプリケーション] ノードに表示されます（ソースは「Citrix Personal vDisk」）。Personal vDiskが有効な場合、これらのどのイベントも表示されません。

イベントのIDが1のものは情報メッセージを意味し、IDが2のものはエラーを意味しています。Personal vDiskのバージョンによっては、一部のイベントが発生しない場合があります。

| イベントID | 説明                                                 |
|--------|----------------------------------------------------|
| 1      | Personal vDisk状態：インベントリの更新を開始しました。                 |
| 1      | Personal vDisk状態：インベントリの更新が完了しました。GUID：%s。         |
| 1      | Personal vDisk状態：イメージ更新を開始しました。                    |
| 1      | Personal vDisk状態：イメージ更新が完了しました。                    |
| 1      | リセット中です。                                           |
| 1      | OK。                                                |
| 2      | Personal vDisk状態：%sにより、インベントリの更新に失敗しました。           |
| 2      | Personal vDisk状態：%sにより、イメージ更新に失敗しました。              |
| 2      | Personal vDisk状態：内部エラーにより、イメージ更新に失敗しました。           |
| 2      | Personal vDisk状態：内部エラーにより、インベントリ更新に失敗しました。         |
| 2      | 正しくシャットダウンされなかったためPersonal vDiskが無効になっています。        |
| 2      | イメージ更新に失敗しました。エラーコードは%dです。                         |
| 2      | Personal vDiskで内部エラーが発生しました。状態コード[%d]、エラーコード[0x%X] |
| 2      | Personal vDiskのリセットに失敗しました。                        |
| 2      | ユーザーのパーソナル設定を格納するためのディスクが見つかりません。                  |
| 2      | このストレージディスクにはPersonal vDiskコンテナの作成に必要な領域がありません。    |

# ユーザープロファイル

Jan 04, 2017

デフォルトでは、マスターイメージ上にVirtual Delivery AgentをインストールするときにCitrix Profile managementが自動的にインストールされます。ただし、プロファイル管理ツールとしてこのコンポーネントを常に使用しなければならないというわけではありません。

ユーザーのニーズに応じてXenApp/XenDesktopポリシーを構成して、各デリバリーグループ内のマシンに異なるプロファイル処理を適用できます。たとえば、あるデリバリーグループではネットワーク上の特定の場所にテンプレートが格納されるCitrix固定プロファイルを使用して、別のデリバリーグループではいくつかのリダイレクトフォルダーと共に別の場所に格納されるCitrix移動プロファイルを使用するポリシーを構成できます。

- 組織内のほかの管理者がXenApp/XenDesktopポリシーを管理する場合は、すべてのデリバリーグループにプロファイル関連のポリシーが正しく適用されるように共同で作業する必要があります。
- Profile managementポリシーは、グループポリシーやProfile managementのINIファイルで設定したり、各仮想マシン上でローカルに設定したりできます。これらの設定は、以下の順に読み取られます。
  1. グループポリシー (ADMまたはADMXファイル)
  2. [ポリシー] ノードのXenApp/XenDesktopポリシー
  3. ユーザーが接続する仮想マシン上のローカルポリシー
  4. Profile managementのINIファイルたとえば、グループポリシーと [ポリシー] ノードの両方で同じポリシーを構成する場合、グループポリシーのポリシー設定が適用され、XenApp/XenDesktopポリシー設定は無視されます。

いずれのプロファイル処理でも、Director管理者はユーザープロファイルの診断情報にアクセスしたりトラブルシューティングを行ったりできます。詳しくは、Directorのドキュメントを参照してください。

Personal vDisk機能を使用する場合、Citrixユーザープロファイルはデフォルトで仮想デスクトップのPersonal vDiskに格納されます。Personal vDiskにコピーが残っている間は、ユーザーストア内のプロファイルのコピーを削除しないでください。これを削除するとProfile managementでエラーが発生し、仮想デスクトップへのログオンに一時プロファイルが使用されることになります。

デスクトップの種類は、インストールされているVirtual Delivery Agentに基づいて自動的に検出され、それに応じてStudioでの構成オプションやProfile managementのデフォルトの動作が設定されます。

Profile managementで設定されるポリシーは、以下の表のとおりです。ポリシーの非デフォルトの設定は保持され、この機能で上書きされることはありません。各ポリシーについて詳しくは、Profile managementのドキュメントを参照してください。プロファイルを作成するマシンの種類により、調整されるポリシーが異なります。最初の要因は、マシンの種類が固定なのかプロビジョニングなのかという点です。次の要因は、それが複数のユーザーによって共有されるのか特定のユーザーに専用のものなのかという点です。

固定システムにはある種のローカル記憶域が備わっていて、システムの電源がオフになってもシステムの内容を維持することができます。固定システムでは、ローカルディスクとして記憶域ネットワーク (SAN) のような記憶域テクノロジーを使用できます。これと対照的に、プロビジョニングシステムは基本ディスクとある種のIDディスクから「オンザフライ」で作成されます。通常、RAMディスクまたはネットワークディスクがローカル記憶域として使用され、ネットワークディスクはしばしば高速リンクのSANによって提供されます。プロビジョニングテクノロジーとは、一般的にProvisioning ServicesまたはMachine Creation Services (またはサードパーティの同等物) を指します。場合により、プロビジョニングされたシステムがPersonal vDiskによって提供される固定ローカルストレージを伴うことがあります。この場合は固定システムとして分類されます。

これらの2つの要因により、以下の種類のマシンが定義されます。

- **固定かつ専用** — Machine Creation Servicesで作成されるデスクトップOSマシンで、Personal vDiskを持ち静的に割り当てられるもの、VDI-in-a-Boxで作成されるデスクトップで、Personal vDiskを持つもの、物理的ワークステーション、およびラップトップコンピューターなど。
- **固定かつ共有** — Machine Creation Servicesで作成されるサーバーOSマシンなど。
- **プロビジョニングかつ専用** — Provisioning Servicesで作成されるデスクトップOSマシンで、Personal vDiskを持たずに静的に割り当てられるものなど。
- **プロビジョニングかつ共有** — Provisioning Servicesで作成されるデスクトップOSマシンで、ランダムに割り当てられるもの、VDI-in-a-Boxで作成されるデスクトップで、Personal vDiskを持たないものなど。

次の表は、各種類のマシンに適したProfile managementポリシー設定を示しています。通常、これらの設定は効果的ですが、必要に応じて変更した方がよい場合もあります。

重要： [ログオフ時にローカルでキャッシュしたプロファイルの削除]、 [プロファイルストリーム配信]、および [常時キャッシュ] は自動構成機能により設定されます。ほかのポリシー設定は、必要に応じて手作業で変更してください。

#### 固定のマシン

| ポリシー                        | 固定かつ専用  | 固定かつ共有  |
|-----------------------------|---------|---------|
| ログオフ時にローカルでキャッシュしたプロファイルの削除 | 無効      | 有効      |
| プロファイルストリーム配信               | 禁止      | 許可      |
| 常時キャッシュ                     | 有効 (注1) | 無効 (注2) |
| アクティブライトバック                 | 無効      | 無効 (注3) |
| ローカル管理者のログオン処理              | 有効      | 無効 (注4) |

#### プロビジョニングされたマシン

| ポリシー                        | プロビジョニングかつ専用 | プロビジョニングかつ共有 |
|-----------------------------|--------------|--------------|
| ログオフ時にローカルでキャッシュしたプロファイルの削除 | 無効 (注5)      | 有効           |
| プロファイルストリーム配信               | 有効           | 有効           |
| 常時キャッシュ                     | 無効 (注6)      | 無効           |
| アクティブライトバック                 | 有効           | 有効           |
| ローカル管理者のログオン処理              | 有効           | 有効 (注7)      |



| ポリシー                                                                                                                                                     | プロビジョニングかつ専用 | プロビジョニングかつ共有 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------------|
| 1. このマシンの種類では [プロファイルストリーム配信] が無効なため、 [常時キャッシュ] 設定は常に無視されます。                                                                                             |              |              |
| 2. [常時キャッシュ] は無効にします。ただし、このポリシー設定を有効にして制限サイズ (MB) を指定すると、ログオン後すぐにサイズの大きなファイルがプロファイルにロードされるようになります。制限サイズ以上のすべてのファイルは、すぐにローカルにキャッシュされます。                   |              |              |
| 3. [アクティブライトバック] は無効にします。ただし、XenAppサーバー間を移動するユーザーのプロファイルの変更を保存する場合は、このポリシー設定を有効にします。                                                                     |              |              |
| 4. [ローカル管理者のログオン処理] は無効にします。ただし、ホスト共有デスクトップの場合は、このポリシー設定を有効にします。                                                                                         |              |              |
| 5. [ログオフ時にローカルでキャッシュしたプロファイルの削除] は無効にします。これにより、ローカルにキャッシュされたプロファイルが保持されます。各マシンが個々のユーザーに割り当てられているため、ログオフ時にマシンがリセットされても、プロファイルのキャッシュによりすばやくログオンできるようになります。 |              |              |
| 6. [常時キャッシュ] は無効にします。ただし、このポリシー設定を有効にして制限サイズ (MB) を指定すると、ログオン後すぐにサイズの大きなファイルがプロファイルにロードされるようになります。制限サイズ以上のすべてのファイルは、すぐにローカルにキャッシュされます。                   |              |              |
| 7. [ローカル管理者のログオン処理] は有効にします。ただし、XenApp/XenDesktopサーバー間を移動するユーザーのプロファイルに対しては、このポリシー設定を無効にします。                                                             |              |              |

フォルダーリダイレクトを有効にすると、ユーザーデータをユーザープロファイルとは異なるネットワーク共有上に格納できます。これにより、プロファイルのサイズが小さくなるため短時間でロードされるようになりますが、ネットワーク帯域幅が消費されます。フォルダーリダイレクト機能では、Citrixユーザープロファイルを使用する必要はありません。管理者は独自にユーザーのプロファイルを管理して、フォルダーをリダイレクトできます。

フォルダーリダイレクトを構成するには、StudioでCitrixポリシーを使用します。

- フォルダーのリダイレクト先のネットワーク共有が使用可能であり、適切なアクセス権が設定されていることを確認します。リダイレクト先のプロパティは自動的に検証されます。
- リダイレクト先のネットワーク共有をセットアップすると、ユーザーの次回ログオン時にプロファイルがリダイレクトされます。

注：フォルダーリダイレクト機能は、CitrixポリシーまたはActive Directoryグループポリシーオブジェクトのいずれか一方を使用して構成してください。両方のポリシーエンジンを使用すると、予期しない問題が発生することがあります。

複数のオペレーティングシステムが混在する展開環境では、ユーザープロファイルの一部がすべてのオペレーティングシステムで共有されるように構成できます。プロファイルの残りの部分は共有されず、単一のオペレーティングシステムでのみ使用されます。異なるオペレーティングシステム上で一貫したユーザーエクスペリエンスを提供するには、オペレーティングシステムごとに異なる構成が必要です。これを詳細なフォルダーリダイレクトと呼びます。たとえば、2つのオペレーティングシステム上で使用される異なるバージョンのアプリケーションで共通のファイルがロードされるようにするには、そのファイルをネットワーク上の単一の場所にリダイレクトします。また、 [スタートメニュー] フォルダーの構造が2つのオペレーティングシステムで異なる場合は、どちらか一方のオペレーティングシステムのフォルダーのみがリダイレクトされるように設定できます。これにより、各オペレーティングシステムでこのフォルダーおよびその内容が分離され、ユーザーに一貫したエクスペリエンスを提供できます。

詳細なフォルダーリダイレクトを使用する場合は、ユーザープロファイル内のデータ構造を理解して、どの部分をオペレーティングシステム間で共有できるかを確認する必要があります。これは、フォルダーリダイレクトによる予期せぬ問題の発生を避けるために重要です。

詳細なフォルダーリダイレクトを使用するには、以下のタスクを行います。

- 各オペレーティングシステムで異なるデリバリーグループを使用します。
- 配信する仮想アプリケーション（仮想デスクトップ上のものを含む）がユーザーのデータや設定をどこに格納するか、およびそのデータ構造を確認します。
- 移動可能な共有プロファイルデータ（異なるオペレーティングシステムでも構造が同じデータ）を含んでいるフォルダーを、各デリバリーグループでリダイレクトされるように設定します。
- 共有できないプロファイルデータについては、1つのデリバリーグループでのみリダイレクトされるように設定します。通常、使用頻度の高いオペレーティングシステムやより実用的なデータのデリバリーグループでリダイレクトを設定します。または、共有できないプロファイルデータを含んでいるフォルダーを、オペレーティングシステムごとに異なるネットワーク共有にリダイレクトすることもできます。

**詳細なフォルダーリダイレクトの例**- この例では、Windows 8とWindows Server 2008で異なるバージョンのMicrosoft OutlookとInternet Explorerがインストールされている場合について説明します。これら2つのオペレーティングシステム用に2つのデリバリーグループをセットアップします。ユーザーがこれらのアプリケーションで共通の「アドレス帳」と「お気に入り」にアクセスできるようにするには、詳細なフォルダーリダイレクトを以下のように構成します。

**重要：**ここで説明する内容は、上記のオペレーティングシステムおよび配信環境での例であり、実際の環境ではさまざまな要因によりフォルダー構造が異なる場合があります。

- これらのデリバリーグループに適用するポリシーで、以下のフォルダーをリダイレクトします。

| フォルダー       | Windows 8でのリダイレクト | Windows Server 2008でのリダイレクト |
|-------------|-------------------|-----------------------------|
| マイドキュメント    | はい                | はい                          |
| アプリケーションデータ | いいえ               | いいえ                         |
| アドレス帳       | はい                | はい                          |
| デスクトップ      | はい                | いいえ                         |
| ダウンロード      | いいえ               | いいえ                         |
| お気に入り       | はい                | はい                          |
| リンク         | はい                | いいえ                         |
| マイミュージック    | はい                | はい                          |
| マイピクチャ      | はい                | はい                          |
| マイビデオ       | はい                | はい                          |
| 検索          | はい                | いいえ                         |
| 保存したゲーム     | いいえ               | いいえ                         |

| フォルダー<br>スタートメニュー | Windows 8でのリダイレクト<br>はい | Windows Server 2008でのリダイレクト<br>いいえ |
|-------------------|-------------------------|------------------------------------|
|                   |                         |                                    |

- オペレーティングシステム間で共有されるフォルダーをリダイレクトする場合、以下の点に注意してください。
  - 「アドレス帳」フォルダーと「お気に入り」フォルダーのリダイレクトを設定する前に、異なるバージョンのOutlookとInternet Explorerでユーザーデータのフォルダー構造を確認してください。
  - 「マイドキュメント」、「マイミュージック」、「マイピクチャ」、および「マイビデオ」の各フォルダーの構造はこれらのオペレーティングシステムで共通なので、両方のデリバリーグループで同じネットワーク共有にリダイレクトできます。
- オペレーティングシステム間で共有できないフォルダーをリダイレクトする場合、以下の点に注意してください。
  - 「デスクトップ」、「リンク」、「検索」、および「スタートメニュー」の各フォルダーの構造はこれらのオペレーティングシステムで異なるため、Windows Server 2008用のデリバリーグループではリダイレクトされないように設定します。これにより、これらのデータは共有されなくなります。
  - 予期せぬ問題の発生を避けるため、これらのフォルダーはWindows 8用のデリバリーグループでのみリダイレクトします。これは、通常の業務ではユーザーがWindows 8を使用することが多く、Windows Server 2008で提供されるアプリケーションには頻繁にアクセスしないためです。また、これらのデータは、アプリケーション環境よりもデスクトップ環境のものの方が実用的です。たとえば、デスクトップ上のショートカットは「デスクトップ」フォルダーに格納されるため、Windows Server 2008マシンよりもWindows 8マシンのデスクトップショートカットをリダイレクトした方が便利です。
- 以下のフォルダーは、オペレーティングシステム間での共有に向いていません。
  - ユーザーがダウンロードしたファイルがサーバー上にコピーされるのを防ぐため、「ダウンロード」フォルダーはリダイレクトしません。
  - 個々のアプリケーションのデータにより互換性やパフォーマンス上の問題が生じることがあるので、「アプリケーションデータ」フォルダーはリダイレクトしません。

フォルダーリダイレクトについて詳しくは、[http://technet.microsoft.com/ja-jp/library/cc766489\(v=ws.10\).aspx](http://technet.microsoft.com/ja-jp/library/cc766489(v=ws.10).aspx)を参照してください。

Citrix Profile managementを使用する場合は、一部のユーザープロファイルフォルダーに対して除外規則を設定して、パフォーマンスを向上できます。この機能を使用する場合は、リダイレクトされるフォルダーに対して除外規則を設定しないでください。フォルダーリダイレクト機能とProfile managementの除外規則を一緒に使用する場合は、リダイレクトされるフォルダーがProfile managementの処理から除外されないようにしてください。これにより、後でリダイレクト機能を無効にしてもユーザープロファイルフォルダー構造の整合性が保持されます。除外について詳しくは、「[項目の包含および除外](#)」を参照してください。

# HDX

Jan 04, 2017

Citrix HDXには、高品位なユーザーエクスペリエンスを提供する、幅広い機能セットが含まれています。

|          |                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| デバイス上    | ユーザーデバイスのコンピューティング能力を利用して、ユーザーエクスペリエンスを拡張および最適化します。HDX MediaStream機能により、スムーズでシームレスなマルチメディアコンテンツが仮想デスクトップやアプリケーションに提供されます。ワークスペースコントロール仮想デスクトップやアプリケーションのセッションを一時停止して、ほかのデバイスでそのセッションでの作業を再開できます。                                                                                                                                                                                                  |
| ネットワーク上  | HDXによる高度な最適化およびアクセラレーションにより、待機時間が長く低帯域幅のWAN接続を含むあらゆるネットワークにおいて最高のパフォーマンスが提供されます。<br>社内ネットワークからデスクトップやアプリケーションにローカルにアクセスする場合やファイアウォールの外側からリモートにアクセスする場合など、環境のさまざまな条件に応じて最適な機能が適用され、パフォーマンスと消費帯域幅のバランスが最適化されます。                                                                                                                                                                                     |
| データセンター内 | HDXでは、サーバー側の処理能力およびスケーラビリティを利用して、クライアントデバイス側の能力に制限されずに高度なグラフィックパフォーマンスを提供できます。圧縮されたマルチメディア情報は、本来の形式のままユーザーデバイスに直接送信されます。<br>Citrix Directorでは、ユーザーデバイスに接続しているHDXチャネルの状態を監視できます。<br>EdgeSight Network InspectorとEdgeSight Performance ManagementをDirectorに統合したものであるHDX Insightは、ICAトラフィックに関するデータを収集して、クライアント側およびサーバー側のICAセッション遅延、ICAチャネルの帯域幅使用量、および各セッションのICA往復時間値などのリアルタイムおよび履歴の詳細をダッシュボードに表示します。 |

仮想デスクトップ上では、以下の操作を行うことでHDXの機能を体験できます。

- See how HDXが仮想デスクトップにどのようにビデオコンテンツを配信するかについては、高品位ビデオを含むWebサイト (<http://www.microsoft.com/silverlight/iis-smooth-streaming/demo/>など) のビデオを参照してください。
- FlashリダイレクトによりFlashマルチメディアコンテンツの配信がどのように高速化されるかを体験するには、次の手順に従います。
  1. Adobe Flash Player (<http://get.adobe.com/flashplayer/>) をダウンロードして仮想デスクトップおよびユーザーデバイスにインストールします。
  2. Desktop Viewerツールバーの [基本設定] をクリックします。[Desktop Viewer基本設定] ダイアログボックスの [Flash] タブで、[コンテンツを最適化する] を選択します。
  3. FlashリダイレクトによるFlashマルチメディアコンテンツの配信パフォーマンスを体験するには、仮想デスクトップでYouTubeなどのWebサイトにアクセスして、Flashビデオを再生します。Flashリダイレクトはシームレスに機能するため、ユーザー側にはいつ動作しているかはわかりません。ただし、Flashリダイレクトが使用される場合、Flash Playerが起動する前に画面上に一瞬だけカラーブロックが現れます。
- HDXにより高品位オーディオがどのように配信されるかを体験するには、次の手順に従います。
  1. Citrix Receiverで、最高の音質を選択します。詳しくは、Receiverのドキュメントを参照してください。
  2. デスクトップ上のデジタルオーディオプレーヤー (iTunesなど) で音楽ファイルを再生します。

HDXでは、特別な構成を行わなくても一般的なユーザーに適したグラフィックおよびビデオ配信が提供されます。Citrixポリ

シー設定は、一般的な使用環境で最適なユーザーエクスペリエンスが提供されるようにデフォルトで有効になっています。

- HDXは、クライアントプラットフォーム、アプリケーション、およびネットワーク帯域幅に基づいて最適な配信方法を自動的に選択し、状況の変化に応じて自動調整します。
- HDXは、2Dおよび3Dのグラフィックおよびビデオのパフォーマンスを最適化します。
- HDXは、すべてのクライアントの仮想デスクトップユーザーにWindows Aeroエクスペリエンスを配信します。
- HDXは、インターネットやイントラネット上のマルチメディアコンテンツなどをホストサーバーを介さず直接ユーザーデバイス上にストリーム配信します。このクライアント側でのコンテンツ取得に必要な条件が満たされない場合、Windows Mediaリダイレクトの設定に基づいてサーバー側からユーザーデバイスにリダイレクトされます。通常、Windows Mediaリダイレクトに関するポリシーを変更する必要はありません。

ヒント：

- HDX機能のサポートおよび要件については、「[XenAppおよびXenDesktop 7.6のシステム要件](#)」を参照してください。特に注記のあるものを除き、WindowsサーバーOSマシン、WindowsデスクトップOSマシン、およびリモートPCアクセスのデスクトップでHDX機能を使用できます。
- このセクションのトピックでは、ユーザーエクスペリエンスをさらに最適化したり、サーバーのスケラビリティを改善したり、消費帯域幅を抑えたりする方法について説明します。Citrixポリシーおよびそのポリシー設定について詳しくは、このリリースの「  
— Citrixポリシー  
」を参照してください。
- レジストリを編集する場合は細心の注意が必要です。レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があります。Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

HDXのデフォルトでは、Windowsデバイスから仮想デスクトップにアクセスするユーザーに、良好なレスポンスのWindows AeroまたはWindows 8デスクトップエクスペリエンスが提供されます。これを可能にするため、ユーザーデバイス側のGPU (Graphics Processing Unit) またはIGP (Integrated Graphics Processor) を使用してローカルのDirectXグラフィック処理を行います。この機能は「デスクトップコンポジションリダイレクト」と呼ばれ、サーバー側の負荷軽減にも役立っています。詳しくは、<http://blogs.citrix.com/2013/11/06/go-supersonic-with-xendesktop-7-x-bandwidth-supercodecs/>の「What to do with all these choices」を参照してください。

ユーザーセッションで消費される帯域幅を抑えるには、以下のポリシー設定の調整を検討してください。ただし、これらの変更によりユーザーエクスペリエンスが低下する場合があります。

- デスクトップコンポジションリダイレクト。WindowsユーザーデバイスからアクセスするWindowsデスクトップOSマシンにのみ適用され、Windowsデスクトップの表示にのみ効果があります。Citrixポリシー設定の「ローカルアプリケーションのアクセスを許可する」で「許可」を指定しても、アプリケーションウィンドウの表示はサーバー側で処理されます。
- デスクトップコンポジションの画質。シームレスアプリケーションまたはローカルアプリケーションアクセスが無効な場合に、デスクトップコンポジションで高品質なグラフィックが使用されます。消費帯域幅を抑えるには、グラフィックの品質を下げます。
- 動的ウィンドウプレビュー。フリップ、フリップ3D、タスクバープレビュー、およびピークウィンドウプレビューモードにおけるシームレスウィンドウの表示の有効/無効を切り替えます。消費帯域幅を抑えるには、このポリシー設定を無効にします。

視覚表示ポリシー設定は、仮想デスクトップからユーザーデバイスに送信されるイメージの品質を制御します。

- 表示品質。ユーザーデバイス上に表示されるイメージの表示品質として、「[低]」、「[中]」、「[高]」、「[常に無損失]」、または「[操作時は低品質]」を指定します。デフォルトは「[中]」です。

- ターゲットフレーム数。仮想デスクトップからユーザーデバイスに送信されるイメージの1秒あたりの最大フレーム数 (fps) を指定します。デフォルトは30fpsです。通常、大きい値を指定することでユーザーエクスペリエンスが向上します。CPUが低速なデバイスでは、小さい値を指定した方がユーザーエクスペリエンスが向上する場合があります。
- 表示メモリの制限。セッションのビデオバッファの最大サイズをキロバイト単位で指定します。デフォルトは65536KBです。高い色数および解像度を使用するセッションでは、大きい値を指定します。必要なメモリの量は算出できません。32ビット以外の色数は、[従来のグラフィックモード] 設定が有効な場合のみ使用できます。

HDX Webカメラビデオ圧縮を使用すると、セッションでのビデオ会議で帯域幅使用効率および遅延に対する耐性が向上します。この機能では、Webカメラのトラフィックが専用のマルチメディア仮想チャネルでストリーム配信されます。HDX Plug-in-Playサポートのアイソクロナス転送に比べて帯域幅消費が少なく、WAN接続に適しています。

このデフォルト設定は、ReceiverユーザーがDesktop Viewerの [マイクとWebカメラ] 設定で [マイクおよびWebカメラを使用しない] を選択することで無効になります。ユーザーがHDX Webカメラビデオ圧縮を無効にすることを禁止するには、ポリシーの [ICA] > [USBデバイス] の設定項目を使ってUSBデバイスのリダイレクトを無効にします。

Receiver for Windowsのデフォルトでは、HDX Webカメラビデオ圧縮が有効になっています。Receiver for Linuxでは、構成の変更が必要です。詳しくは、Receiverのドキュメントを参照してください。HDX Webカメラビデオ圧縮を使用するには、以下のポリシー設定を有効にする必要があります (これらの設定項目はデフォルトで有効になっています)。

- クライアントオーディオリダイレクト
- クライアントマイクリダイレクト
- マルチメディア会議
- Windowsメディアリダイレクト

WebカメラでH.264ハードウェアエンコード機能がサポートされる場合、HDX Webカメラビデオ圧縮ではデフォルトでそのハードウェアエンコードが使用されます。ハードウェアエンコードでは追加の帯域幅が使用されるため、狭帯域幅接続には適していません。低帯域幅ネットワークでソフトウェアエンコードが使用されるようにするには、レジストリキー HKEY\_CURRENT\_USER\Software\Citrix\HdxRealTime に DWORD 値 DeepCompress\_ForceSWEncode=1 を設定します。

ユーザーエクスペリエンスの向上よりもサーバースケーラビリティの改善が重要な環境では、従来のグラフィックモードが採用されるように設定します。これを行うには、ポリシーの [従来のグラフィックモード] 設定を追加して、個々のグラフィックポリシー設定を構成します。従来のグラフィックモードを使用すると、WANやモバイル接続時のユーザーエクスペリエンスが低下することがあります。

# Thinwire互換モード

Jan 04, 2017

このアーティクルは7.6 FP3に適用されます。

Thinwire互換モードは、エンドユーザーエクスペリエンスを犠牲にすることなく帯域幅の使用を抑えて高度なサーバースケーラビリティを実現する新しい画面分解およびキャッシュテクノロジーを使用します。

Thinwire互換モードには、次の機能があります。

- ビットマップ専用プロバイダー用のインテリジェントビットマップ一致。
  - ウィンドウ移動およびスクロールを効率的にするためのビットマップ変換解析です。
- 後方互換性。クライアントまたはCitrix Receiverのアップグレード、あるいはハードウェアアクセラレーションに対する要件はありません。
  - テストは、最新のものから5年を越えるのものまでの古いシンクライアントで実行されています。
- サーバーCPUの使用をととも低くするために最適化、およびサーバースケーラビリティの改善。
- 通常のワークロードについて帯域幅をさらに15~20%に減少するエミュレートされた16ビットモード。
- サーバー側でレンダリングするビデオコンテンツの遷移の検出。
  - 改善されたマルチメディアエクスペリエンスのためのマルチ遷移の処理。たとえば、複数のビデオまたは紙テープを見る場合。
  - 遷移状態のままにする地域を選択的にシャープにします。
- CloudBridgeアクセラレーションの最適化。テストでは、Officeタイプのワークロードについて、帯域幅の減少の比率は最大で6:1となりました。
- ポリシー設定を介して有効にできるアダプティブ表示。詳しくは、[動画のポリシー設定](#)の「[動画の圧縮](#)」を参照してください。
- Windows 10を含むVDAとWindowsをサポート。
- 低帯域幅でのユーザーエクスペリエンスをより良いものにするため応答性、相互作用性、および割り込み可能なシャープ化を向上させる新しい3D Proの「[操作時は低品質](#)」モード。
- 従来のグラフィックモードよりも高い、デフォルトの静的写真画像品質。

表示品質設定の「低」、「中」（デフォルト）および「高」については、アダプティブ表示ポリシーに従って、遷移検出機能が動的に画面の更新を評価し高度にアニメ化された領域を低品質にして送信する必要があるかどうかを判別され、クライアントのパフォーマンスを向上させ、帯域幅の使用を減少させます。

【[操作時は低品質](#)】の画質については、Thinwire互換モードでは大画面の更新に対して「ファジーファースト」アプローチが用いられます。この設定は、3Dモデルまたはそのほかのグラフィック指向アプリケーションを操作する3D Proユーザーを対象にしたものです。アクティビティが続行する場合には、引き続き遷移的モードとして処理され、遷移的アクティビティが準止すると影響を受ける領域がシャープになりキャッシュされます。たとえばワイヤーフレームを回転するなど初期の大きな変更については、変更を受ける領域である種の簡易イメージ解析が実行され、「ファジー遷移性」を前提とするか、それとも「シャープ遷移性」（無損失）を前提とするかが判断されます。FPS（1秒あたりのフレーム数）および帯域幅については、Citrix無損失コーデックを使って単純なイメージをエンコードすることがより効率的であり、画質も損なわれません。

操作時は低品質のシャープ化から無損失のステップもまた異なります。影響のある領域が一度にシャープになるのではなく、その部分は事前決定されているブロックでシャープ化されるため、相互作用性と快適なユーザーエクスペリエンスを維持するのに役立ちます。遷移中に大きな変更領域をシャープにすると、特に低帯域幅回線においてたとえば3Dモデルの移動を一時的に停止してから再度実行することにより以前は「詰まり」を引き起こしていました。シャープ化するブロックのサイズは、ターゲットの最小フレームレート（アダプティブ表示ポリシー設定）を維持しようとするためにどのくらい品質を低下させかにより異なります。品質を著しく低下させた場合、シャープにするブロックサイズはより小さく、最小で128 x 128ピクセル

ルとなります。たとえばクライアントに十分な処理能力と帯域幅があるなど品質をそれほど低下させない場合は、シャープにするブロックサイズは最大で384 x 384ピクセルにできます。



# HDX 3D Pro

Jan 04, 2017

HDX 3D Proを使用すると、OpenGLやDirectXテクノロジーを使用する専門的な3Dグラフィック処理アプリケーションなど、GPU (Graphics Processing Unit) でのハードウェアアクセラレーションに適したデスクトップおよびアプリケーションを配信できます (標準VDAでは、DirectXのGPUアクセラレーションのみがサポートされます)。

これらの3Dグラフィック処理アプリケーションとして次のものがあります。

- コンピューター支援設計 (CAD)、コンピューター支援製造 (CAM)、およびコンピューター支援エンジニアリング (CAE) アプリケーション
- 地理情報システム (GIS) ソフトウェア
- 医療画像処理のための画像保存通信システム (PACS)
- 最新バージョンのOpenGL、DirectX、NVIDIA CUDA、およびOpenCLを使用するアプリケーション
- 並列計算にNVIDIA Compute Unified Device Architecture (CUDA) GPUを使用する計算集約型の非グラフィックアプリケーション

HDX 3D Proでは、さまざまな帯域幅において最適なユーザーエクスペリエンスが提供されます。

- ワイドエリアネットワーク (WAN) 接続の場合：帯域幅が1.5MbpsのWAN接続でもインタラクティブなユーザーエクスペリエンスが提供されます。
- ローカルエリアネットワーク (LAN) 接続の場合：帯域幅が100MbpsのLAN接続ではローカルデスクトップに匹敵するユーザーエクスペリエンスが提供されます。  
ユーザーが使用する複雑で高価なワークステーションをよりシンプルなユーザーデバイスに置き換えて、グラフィック処理をユーザー側から中央管理が可能なデータセンター内に移管できます。

HDX 3D Proにより、WindowsデスクトップOSマシンとWindowsサーバーOSマシンでのGPUアクセラレーションが提供されます。また、Citrix XenServerおよびNVIDIA社のGRID GPUを使用する環境では、WindowsデスクトップOSマシンで仮想GPU (vGPU) アクセラレーション機能を使用できます。サポートされるXenServerのバージョンについては、「[Citrix Virtual GPU Solution](#)」を参照してください。

Health Checkツールの後継であるHDX Monitorツールを使用すると、HDX仮想テクノロジーの操作と構成を検証して、HDXの問題を診断して解決できます。このツールの詳細およびダウンロードについては、<https://taas.citrix.com/hdx/download/>を参照してください。

# Flashリダイレクト

Jan 04, 2017

Flashリダイレクト機能を有効にすると、ほとんどのAdobe Flashコンテンツ（アニメーション、ビデオ、アプリケーションなど）がLANまたはWANで接続されたユーザー側のWindowsデバイス上で処理（レンダリング）され、サーバーリソースおよびネットワークの負荷が軽減されます。これにより、スケーラビリティおよびユーザーエクスペリエンスが向上します。Flashダイレクトを構成するには、サーバー側およびクライアント側の設定が必要です。

注意：Flashリダイレクトでは、ユーザーデバイスとサーバー間で頻繁な相互通信が行われます。ユーザーデバイスとサーバー間でセキュリティ境界による分離が不要な環境でのみ、この機能を使用してください。また、ユーザーデバイスでは、信頼するサーバーでのみこの機能を使用するように設定する必要があります。Flashリダイレクトを使用するにはユーザーデバイスにFlash Playerをインストールする必要があるため、ユーザーデバイス上のFlash Playerの安全性が確認されている場合のみFlashリダイレクトを有効にしてください。

第2世代および従来のバージョンのFlashリダイレクトは個別のソリューションであり、異なる仮想チャネルで実行されます。

- 従来のバージョンのFlashリダイレクトは、クライアント側でのみサポートされます。ユーザーデバイスに以前のバージョンのFlash Playerがインストールされている、またはユーザーデバイスにFlash Playerがインストールされていない場合、Flashコンテンツはサーバー側でレンダリングされます。
- 第2世代のFlashリダイレクトは、クライアント側およびサーバー側でサポートされます。クライアント側で第2世代のFlashリダイレクトがサポートされる場合、Flashコンテンツはクライアント側でレンダリングされます。第2世代のFlashリダイレクト機能では、以下で説明するWAN接続、インテリジェントフォールバック、URL互換性リストがサポートされます。

Flashリダイレクトは、サーバー上のWindowsイベントログを使用してFlashイベントのログを記録します。このイベントログには、Flashリダイレクトが使用されたかどうか、および問題に関する詳細が記録されます。Flashリダイレクトのすべてのイベントで、以下の共通の情報が記録されます。

- Flashリダイレクトのイベントは、アプリケーションログに記録されます。
- Windows 7またはWindows 8が動作するコンピューターでは、Flashリダイレクトのログは [アプリケーションとサービス ログ] ノードに記録されます。
- ソースには「Flash」と記録されます。
- 分類には「なし」と記録されます。

HDX Flashの互換性の最新情報については、[CTX136588](#)を参照してください。

サーバー側でFlashリダイレクト機能を構成するには、Citrixポリシーで以下の設定項目を使用します。詳しくは、[ファイルリダイレクトのポリシー設定](#)を参照してください。

- [Flashアクセラレーションのデフォルトの動作] 設定を使用して、Flashアクセラレーションのデフォルトの動作を設定します。デフォルトでは、Flashリダイレクトは有効になっています。特定のWebページのFlashコンテンツでこのデフォルト設定を上書きするには、[Flash URL互換性リスト] 設定を使用します。
- [Flashインテリジェントフォールバック] 設定により、小さいFlashムービー（広告で 사용되는ものなど）が検出され、それらがユーザーデバイスではなくサーバー側でレンダリングされます。この機能を使用しても、WebページやFlashアプリケーションのロード時に中断や障害は発生しませんデフォルトでは、Flashインテリジェントフォールバックは有効になっています。サイズにかかわらずすべてのFlashコンテンツがユーザーデバイス側でレンダリングされるようにするには、この設定項目を無効にします。
- [Flashサーバー側でのコンテンツ取得URLリスト] 設定を使用すると、特定のWebサイトのFlashコンテンツがいったんサーバー上にダウンロードされ、それがユーザーデバイスに転送されてレンダリングされるようになります。Flashリダイレクトのデフォルトでは、Flashコンテンツがユーザーデバイス上にダウンロードされ、そこでレンダリングされます。この設定項目を使用する場合はユーザーデバイス側の [サーバー側のコンテンツの取得を有効にする] 設定が必要で、イントラネット上の内部用Flashアプリケーションで使用されます。詳しくは、後述の説明を参照してください。この機能はほと

多くのインターネットサイトでも動作するため、ユーザーデバイスが直接インターネットに接続できない環境 (XenApp/XenDesktopサーバー経由でインターネットに接続している場合など) で使用できます。

注：サーバー側でのコンテンツ取得機能は、RTMP (Real Time Messaging Protocol) を使用したFlashアプリケーションをサポートしません。この場合、サーバー側でのレンダリングが使用され、HTTPおよびHTTPSがサポートされます。

- [Flash URL互換性リスト] 設定では、特定のWebサイト上のFlashコンテンツをクライアント側でレンダリングするか、サーバー側でレンダリングするか、またはブロックするかを指定します。
- [Flash背景色リスト] 設定では、WebページとFlashインスタンスの表示色を一致させて、Flashリダイレクトを使用しているときのWebページの表示を改善させることができます。

ユーザーデバイス側でFlashリダイレクト機能を使用するには、Citrix Receiverに加えてAdobe Flash Playerをインストールする必要があります。ユーザーデバイス上でこれ以外の特別な構成は必要ありません。

デフォルトの設定は、Active Directoryグループポリシーオブジェクトを使用して変更できます。HDX MediaStream Flashリダイレクトのクライアント管理テンプレート (HdxFlashClient.adm) をインポートして追加します。このテンプレートは、以下の場所からインポートできます。

- 32ビットコンピューター：%Program Files%\Citrix\ICA Client\Configuration\ja
- 64ビットコンピューター：%Program Files (x86)%\Citrix\ICA Client\Configuration\ja

[管理用テンプレート]、[従来の管理用テンプレート (ADM)]、[HDX MediaStream Flashリダイレクト - クライアント] の順に開きます。Active Directoryグループポリシーオブジェクトおよびテンプレートについて詳しくは、Microsoft社のドキュメントを参照してください。

## Flashリダイレクトの動作を変更する

Adobe Flashコンテンツをユーザーデバイス側にリダイレクトしてローカルで処理されるようにするには、サーバー側での設定に加えて、[このユーザーデバイスでのHDX MediaStream Flashリダイレクトを有効にする]ポリシーを設定します。Flashリダイレクトはデフォルトで有効になっており、検出されたネットワークの状態に応じて自動的にユーザーデバイス上でFlashコンテンツが再生されます (インテリジェントネットワーク検出)。

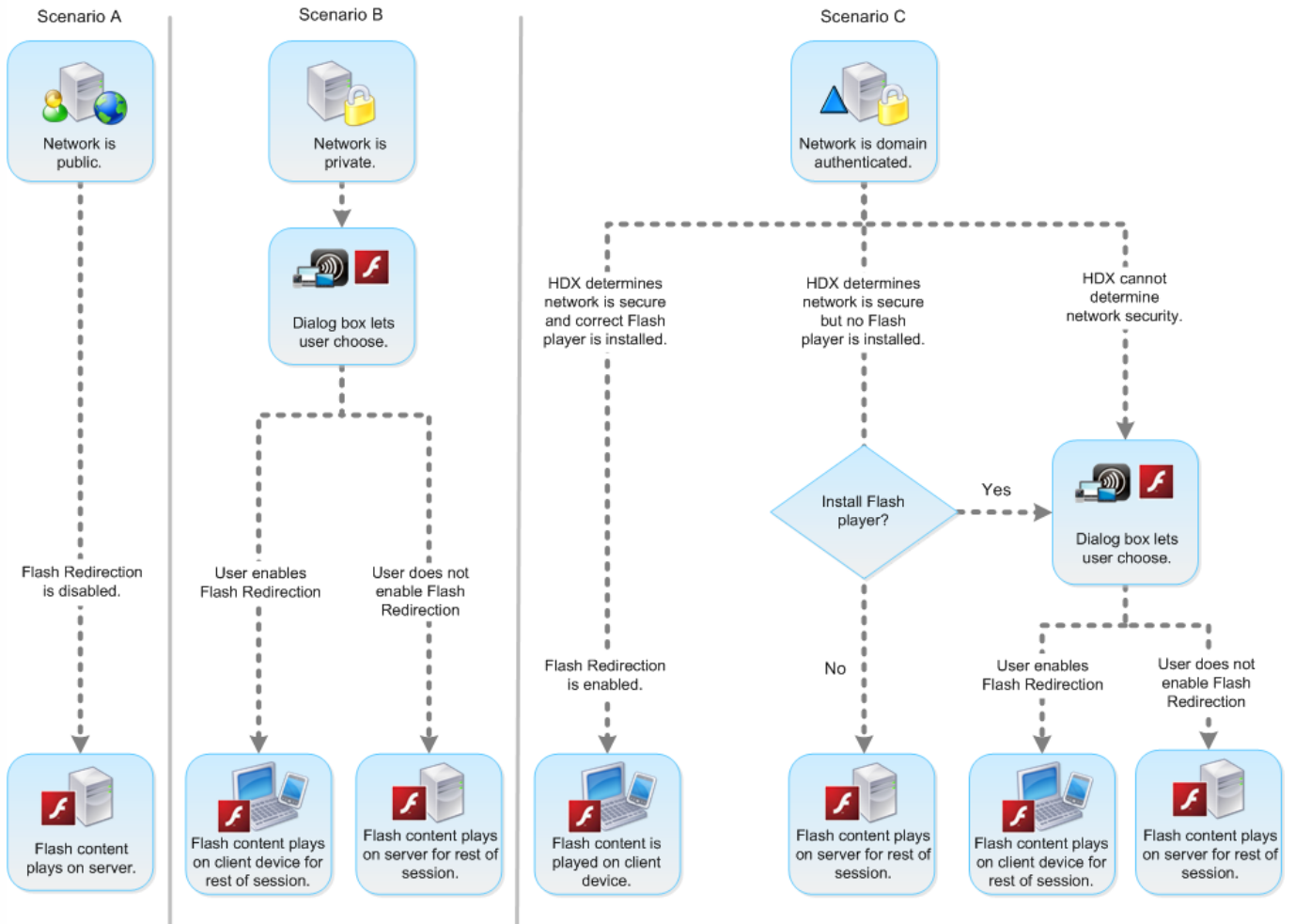
ポリシー設定が未構成である場合、Desktop Lock環境ではデフォルトでFlashリダイレクトが有効になります。

Flashリダイレクトの動作を変更したり、リダイレクト機能を無効にしたりするには、以下の手順に従います。

1. [設定] の一覧で [このユーザーデバイスでのHDX MediaStream Flashリダイレクトを有効にする] を選択して [ポリシー設定] をクリックします。
2. [未構成]、[有効] (デフォルト)、または [無効] を選択します。
3. [有効] を選択する場合は、[HDX MediaStream Flashリダイレクトを使用する] 一覧で以下のいずれかのオプションを選択します。
  - 最新のFlashリダイレクト機能 (第2世代のFlashリダイレクト) を使用するには、[第2の生成でのみ有効] (第2世代Flashリダイレクトのこと) を選択します。これにより、要件が満たされる場合はFlashコンテンツがユーザーデバイス側で処理され、満たされない場合はサーバー側で処理されます。
  - 常にFlashリダイレクトを行うには、[常に有効] を選択します。これにより、Flashコンテンツが常にユーザーデバイス側で処理されます。
  - Flashリダイレクトを無効にするには、[常に無効] を選択します。これにより、Flashコンテンツが常にサーバー側で処理されます。
  - クライアント側のネットワークのセキュリティレベルを検出してFlashリダイレクトの使用を決定するには、[確認] (デフォルト) を選択します。ネットワークのセキュリティレベルを検出できない場合は、Flashリダイレクト機能を使用するかどうかを確認するメッセージがユーザーに表示されます。ネットワークのセキュリティレベルを検出できない場合は、Flashリダイレクト機能を使用するかどうかを確認するメッセージがユーザーに表示されます。

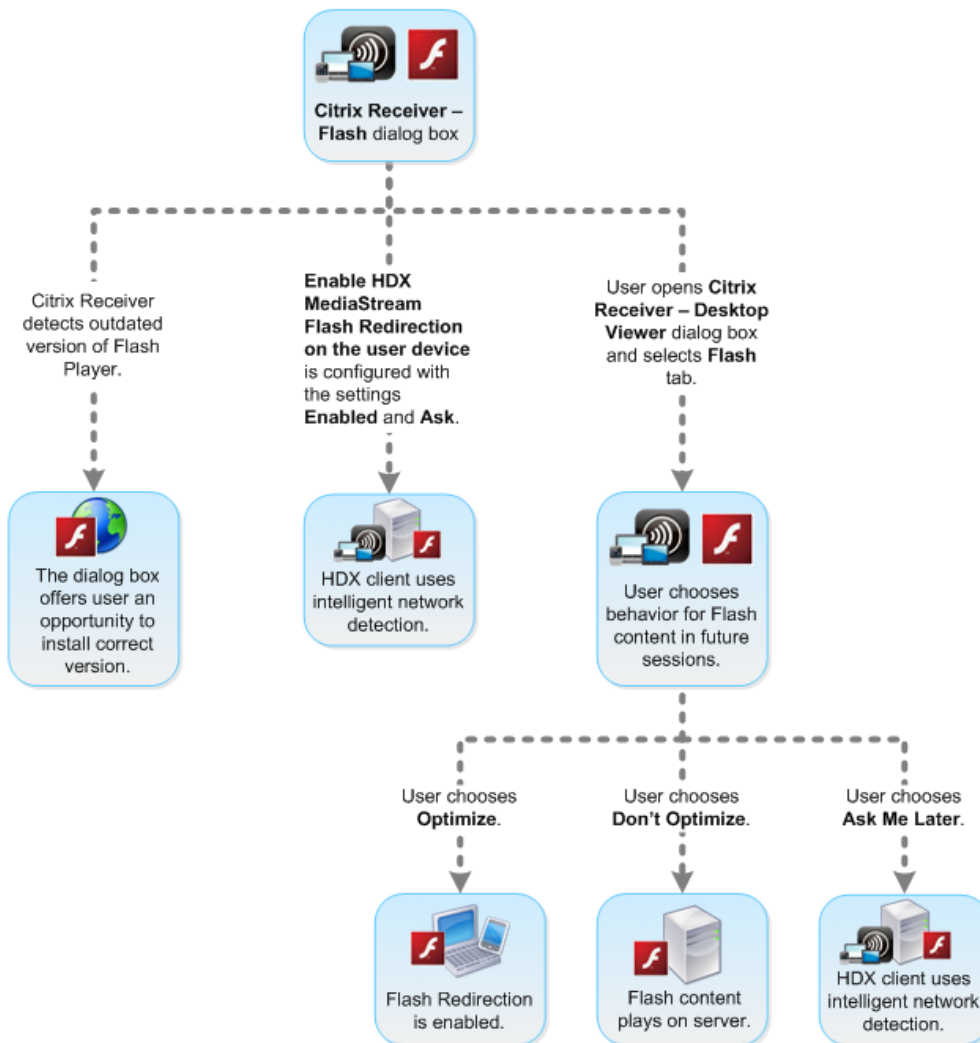
次の図は、さまざまなネットワークでFlashリダイレクトがどのように処理されるかを示しています。

### Intelligent Network Detection for Flash Redirection



インテリジェントネットワーク検出機能は、ユーザー側で有効または無効にすることができます。これを行うには、[Citrix Receiver - Desktop Viewer基本設定] ダイアログボックスの [Flash] タブで、[最適化する] または [最適化しない] を選択します。ここに表示されるオプションは、次の図のように、ユーザーデバイスでのFlashリダイレクトの構成により異なります。

## User control of Flash redirection



### クライアント側とサーバー側のHTTP Cookieを同期する

クライアント側とサーバー側のHTTP Cookieの同期機能は、デフォルトで無効になっています。この同期機能を有効にすると、サーバー側からクライアント側にHTTP Cookieがダウンロードされます。ダウンロードされたHTTP Cookieはクライアント側でのコンテンツ取得に使用され、そのFlashコンテンツのWebサイトがそのCookieを必要に応じて使用できるようになります。

注：この同期処理が発生しても、クライアント側のCookieは上書きされません。後で同期機能が無効になった場合は、元のクライアント側のCookieが使用されます。

1. [設定] の一覧で [クライアント側のHTTP Cookieとサーバー側のCookieの同期化を有効にする] を選択し、[ポリシー設定] をクリックします。
2. [未構成]、[有効]、または[無効]（デフォルト）を選択します。

### サーバー側でのコンテンツ取得を有効にする

Flashリダイレクトのデフォルトでは、Adobe Flashコンテンツがユーザーデバイス上にダウンロードされ、そこで処理されます。サーバー側でのコンテンツ取得を有効にすると、Flashコンテンツがサーバー上にダウンロードされ、その後でユーザーデバイスに転送されます。[Flash URL互換性リスト] 設定によりブロックされるサイトなどの上書きポリシーがない限り、Flashコンテンツはユーザーデバイス側で処理されます。

通常、サーバー側でのコンテンツ取得は、ユーザーデバイスがNetScaler Gatewayを介して社内サイトに接続する場合や、ユーザーデバイスがインターネットに直接アクセスできない場合に使用されます。

注：サーバー側でのコンテンツ取得機能は、RTMP（Real Time Messaging Protocol）を使用したFlashアプリケーションをサポートしません。このようなサイトは、サーバー側で処理されます。

第2世代のFlashリダイレクトでは、サーバー側でのコンテンツ取得を有効にする3つのオプションとがサポートされています。これらの中には、サーバー側で取得したコンテンツをユーザーデバイス上にキャッシュするためのものが含まれています。これにより、再使用されるコンテンツがユーザーデバイス上にキャッシュされるため、パフォーマンスが向上します。このキャッシュコンテンツは、ユーザーデバイス上でキャッシュされるほかのHTTPコンテンツとは別に保持されます。

第2世代のFlashリダイレクトでは、いずれかの有効オプションが選択されている場合に、クライアント側でのSWFファイル取得に失敗すると自動的にサーバー側でのコンテンツ取得にフォールバックされます。

サーバー側でのコンテンツ取得を有効にするには、クライアントデバイス側およびサーバー側での設定が必要です。

1. [設定] の一覧で、[サーバー側のコンテンツの取得を有効にする] を選択して [ポリシー設定] をクリックします。
2. [未構成]、[有効]、または[無効]（デフォルト）を選択します。[有効] を選択する場合は、[サーバー側のコンテンツの取得の状態] 一覧で以下のいずれかのオプションを選択します。

| オプション                   | 説明                                                                                                                                                                                 |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 無効                      | サーバー側でのコンテンツ取得を無効にします。サーバー上の [Flashサーバー側でのコンテンツ取得 URL リスト] 設定は無視されます。サーバー側でのコンテンツ取得へのフォールバックも無効になります。                                                                              |
| 有効                      | [Flashサーバー側でのコンテンツ取得 URL リスト] で指定した Web ページおよび Flash アプリケーションに対するサーバー側でのコンテンツ取得を有効にします。サーバー側でのコンテンツ取得へのフォールバックは使用可能になりますが、Flash コンテンツはキャッシュされません。                                  |
| 有効<br>(永続<br>キャッ<br>シュ) | [Flashサーバー側でのコンテンツ取得 URL リスト] で指定した Web ページおよび Flash アプリケーションに対するサーバー側でのコンテンツ取得を有効にします。サーバー側でのコンテンツ取得へのフォールバックが使用可能になります。サーバー側で取得されたコンテンツはユーザーデバイス上でキャッシュされ、そのセッションを終了しても保持されます。   |
| 有効<br>(一時<br>キャッ<br>シュ) | [Flashサーバー側でのコンテンツ取得 URL リスト] で指定した Web ページおよび Flash アプリケーションに対するサーバー側でのコンテンツ取得を有効にします。サーバー側でのコンテンツ取得へのフォールバックが使用可能になります。サーバー側で取得されたコンテンツはユーザーデバイス上でキャッシュされますが、そのセッションの終了時に削除されます。 |

3. サーバー上で、ポリシーの [Flashサーバー側でのコンテンツ取得 URL リスト] 設定を有効にして、URL を一覧に追加します。

クライアント側でのコンテンツ取得でユーザーデバイスをほかの Web サイトにリダイレクトする

Flash コンテンツの取得をリダイレクトするには、第2世代のFlashリダイレクト機能である [クライアント側のコンテンツの取得の URL 書き換え規則] 設定を使用します。この機能を構成する場合、2つの URL のパターンを指定します。ユーザーデバイスが最初のパターン（「一致パターン」）と一致する Web サイトからコンテンツを取得しようとする時、2つ目のパターン（「置換パターン」）により指定された Web サイトにリダイレクトされます。

この設定を使って、コンテンツ配信ネットワーク（CDN）の機能を補うことができます。Flash コンテンツを配信する一部の Web サイトでは、ユーザーからのコンテンツ要求が、そのユーザーに最も近い Web サイトにリダイレクトされま

す。これを、「CDNリダイレクト」と呼びます。Flashリダイレクトによるクライアント側でのコンテンツ取得を使用する場合、Flashコンテンツはユーザーデバイスから要求され、それ以外のWebページの内容はサーバーから要求されます。CDNを使用しているWebサイトでは、サーバーからの要求が最も近いWebサイトにリダイレクトされるため、ユーザーデバイスからの要求も同じ場所にリダイレクトされます。ただし、この場所はユーザーデバイスに最も近い場所ではない可能性があります。このため、WebページとFlashコンテンツの読み込みに距離による差が生じることがあります。

1. [設定] の一覧で、[クライアント側のコンテンツの取得のURL書き換え規則] を選択して [ポリシー設定] をクリックします。
2. [未構成]、[有効]、または[無効] を選択します。デフォルトでは[未構成] が選択されています。[無効] を選択すると、次の手順で指定するURL書き換え規則がすべて無視されます。
3. この設定を有効にした場合は、[表示] をクリックします。Perl正規表現構文を使って [値の名前] ボックスに一致パターンを入力し、[値] ボックスに置換パターンを入力します。

XenAppおよびXenDesktop 7.6 FP3では、Receiver for WindowsまたはReceiver for Linuxを使ってVDAにアクセスするクライアントデバイスに対するFlashリダイレクトに必要な最小バージョンを指定してレジストリ設定を追加できます。

## 警告

レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

**ServerFlashPlayerVersionMinimum**は、ICAサーバー (VDA) 上で必要なFlash Playerの最小バージョンを指定する文字列値です。

**ClientFlashPlayerVersionMinimum**は、ICAクライアント (Citrix Receiver) 上で必要なFlash Playerの最小バージョンを指定する文字列値です。

これらのバージョン文字列は、「10」、「10.2」、または「10.2.140」のように指定できます。現在、メジャーおよびマイナーのビルド番号のみ比較されます。レビジョン番号は無視されます。たとえば、メジャー番号のみ「10」と指定されたバージョン文字列の場合、マイナーおよびビルド番号は0であるという前提になります。

**FlashPlayerVersionComparisonMask**は、DWORD値です。0と設定すると、ICAサーバー上のFlash Playerに対するICAクライアント上のFlash Playerのバージョンの比較が無効になります。比較マスクにはほかの値がありますが、0以外のマスクの意味は変わることがあるため、それらを使用するべきではありません。必要なクライアントに対して比較マスクを0に設定することを推奨します。クライアント不明な設定について比較マスクを設定することはお勧めしません。比較マスクを指定しない場合、Flashリダイレクトを実行するにはICAクライアントにICAサーバー上のFlash Playerのバージョンと同じまたはそれ以上のバージョンのFlash Playerがある必要があります。この場合、Flash Playerのメジャーバージョン番号のみが比較の対象となります。

リダイレクトを実行するには、比較マスクを使ったチェックに加えてクライアントとサーバーの最小チェックにも成功する必要があります。

サブキーClientID0x51はLinux ICAクライアントを指定します。サブキーClientID0x1はWindows ICAクライアントを指定します。このサブキーには、文字列"ClientID"に16進数のクライアント製品ID (0で始まるものがない) を付けて名前が付けられません。クライアントIDの完全な一覧は、Mobile SDK for Windows Appsドキュメント (<http://www.citrix.com/mobilitysdk/docs/clientdetection.html>) で確認できます。

### 32ビットVDAのレジストリ構成例

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\HdxMediaStreamForFlash\Server\PseudoServer] クライアント不明の設定

"ClientFlashPlayerVersionMinimum"="13.0" ICAクライアントに必要な最小バージョン

"ServerFlashPlayerVersionMinimum"="13.0" ICAサーバーに必要な最小バージョン

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ClientID0x1] Windows ICAクライアント設定

"ClientFlashPlayerVersionMinimum"="16.0.0" Windowsクライアントに必要なFlash Playerの最小バージョンを指定します

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ClientID0x51] Linux ICAクライアント設定

"FlashPlayerVersionComparisonMask"=dword:00000000 Linuxクライアントのバージョン比較検査を無効にします (クライアントにサーバーよりも新しいFlash Playerがあるか確認します) "ClientFlashPlayerVersionMinimum"="11.2.0" LinuxクライアントのFlash Playerの最小バージョンを指定します。

### 64ビットVDAのレジストリ構成例

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer]

"ClientFlashPlayerVersionMinimum"="13.0" "ServerFlashPlayerVersionMinimum"="13.0"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ClientID0x1]

"ClientFlashPlayerVersionMinimum"="16.0.0"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\HdxMediaStreamForFlash\Server\PseudoServer\ClientID0x51]

"FlashPlayerVersionComparisonMask"=dword:00000000 "ClientFlashPlayerVersionMinimum"="11.2.0"



# ホストからクライアントへのリダイレクト

Jan 04, 2017

コンテンツのリダイレクト機能では、ユーザーが特定の種類のコンテンツにアクセスするときに、公開アプリケーションを使うのか、ユーザーデバイス上のアプリケーションを使うのかを制御できます。

ホストからクライアントへのリダイレクトは、コンテンツリダイレクト機能の一種です。この機能は、サーバーOSのVDAでのみサポートされ、デスクトップOSのVDAではサポートされません。

- ホストからクライアントへのリダイレクト機能を有効にすると、URLはサーバーVDAでインターセプトされてからユーザーデバイスに送信されます。これらのURLは、ユーザーデバイス上のWebブラウザまたはマルチメディアプレーヤーで開かれます。
- ホストからクライアントへのリダイレクト機能を有効にしても、ユーザーデバイスからURLに接続できない場合は、そのURLがサーバーVDAに戻されます。
- ホストからクライアントへのリダイレクト機能が無効な場合、URLはサーバーVDA上のWebブラウザまたはマルチメディアプレーヤーで開きます。
- ホストからクライアントへのリダイレクト機能が有効な場合、ユーザーがこの機能を無効にすることはできません。

ホストからクライアントへのリダイレクト機能は、以前は**サーバーからクライアントへのリダイレクト機能**と呼ばれていました。

一般的ではない特定の状況では、パフォーマンス、互換性、コンプライアンスなどの理由から、ホストからクライアントへのリダイレクト機能の使用が必要となることがあります。通常は、ほかのコンテンツリダイレクト機能を使用することをお勧めします。

## 性能は、

ホストからクライアントへのリダイレクト機能を使用すると、ユーザーデバイスにアプリケーションがインストールされていれば、VDA上のアプリケーションではなくこのアプリケーションを使用することができ、パフォーマンスを改善できます。

ただし、VDAではAdobe Flashなどのマルチメディアコンテンツは既に最適化されているため、ホストからクライアントへのリダイレクト機能によりパフォーマンスが改善されるのは特定の状況に限られます。まず、ホストからクライアントへのリダイレクト機能ではなく、次の表に示した他のアプローチ（ポリシー設定）を使用することを検討してください。これらのアプローチは柔軟性が高く、特に低性能のユーザーデバイスでは、一般的にユーザーエクスペリエンスも向上します。

## 互換性

以下の場合、ホストからクライアントへのリダイレクト機能を使用して互換性を確保できます。

- HTMLおよびマルチメディア以外のコンテンツタイプを使用する場合（例：カスタムURLタイプ）
- VDAのマルチメディアプレーヤーのマルチメディアリダイレクト機能ではサポートされていない旧式のメディア形式を使用する場合（Real Mediaなど）
- 特定のコンテンツタイプのアプリケーションを使用するユーザーがごく少数であり、それらのユーザーがこのアプリケーションを各ユーザーデバイスにインストール済みの場合
- VDAでは特定のWebサイトにアクセスできない場合（例：内部Webサイトから別組織のWebサイトへのアクセス）

## コンプライアンス

以下の場合、ホストからクライアントへのリダイレクト機能を使用してコンプライアンスを確保できます。

- アプリケーションまたはコンテンツのライセンス契約によりVDA経由の公開が許可されていない場合
- 組織のポリシーによりVDAへのドキュメントのアップロードが許可されていない場合

一部の状況は、環境が複雑な場合や、ユーザーデバイスとVDAを所有する組織が異なる場合に起こりやすくなります。

環境によっては、さまざまな種類のユーザーデバイスが存在する場合があります。

| ユーザーデバイス      | 状況または環境                                                       | 使用するコンテンツリダイレクト機能           |
|---------------|---------------------------------------------------------------|-----------------------------|
| タブレット         | -                                                             | 次表のすべての機能                   |
| ラップトップPC      | -                                                             | 次表のすべての機能                   |
| デスクトップPC      | ユーザーがユーザーデバイスにインストール済みの多様なアプリケーションを使用する場合                     | 次表のすべての機能                   |
| デスクトップPC      | ユーザーが使用するのが、ユーザーデバイスにインストール済みである少数かつ既知のアプリケーションに限られている場合      | ローカルアプリケーションアクセス            |
| デスクトップPC      | ユーザーがユーザーデバイスにインストール済みのアプリケーションを使用しない場合                       | マルチメディアリダイレクトまたはFlashリダイレクト |
| デスクトップアプライアンス | ベンダーがマルチメディアリダイレクトまたはFlashリダイレクトをサポートしている場合                   | マルチメディアリダイレクトまたはFlashリダイレクト |
| シンクライアント      | ベンダーがマルチメディアリダイレクト、Flashリダイレクト、ホストからクライアントへのリダイレクトをサポートしている場合 | 次表のすべての機能                   |
| ゼロクライアント      | ベンダーがマルチメディアリダイレクトまたはFlashリダイレクトをサポートしている場合                   | マルチメディアリダイレクトまたはFlashリダイレクト |

使用するコンテンツリダイレクト機能を決める際は、次の例を参考にしてください。

| URLのリンク         | 状況または環境            | 使用するコンテンツリダイレクト機能   |
|-----------------|--------------------|---------------------|
| Webページまたはドキュメント | VDAがURLにアクセスできない場合 | ホストからクライアントへのリダイレクト |

|                     |                                          |                     |
|---------------------|------------------------------------------|---------------------|
| Webページ              | WebページにAdobe Flashが含まれる場合                | Flashリダイレクト         |
| マルチメディアファイルまたはストリーム | 互換性のあるマルチメディアプレーヤーがVDAにインストールされている場合     | マルチメディアリダイレクト       |
| マルチメディアファイルまたはストリーム | 互換性のあるマルチメディアプレーヤーがVDAにインストールされていない場合    | ホストからクライアントへのリダイレクト |
| ドキュメント              | VDAに当該種類のドキュメント用のアプリケーションがインストールされていない場合 | ホストからクライアントへのリダイレクト |
| ドキュメント              | ユーザーデバイスへのドキュメントのダウンロードが禁止されている場合        | リダイレクト不可            |
| ドキュメント              | VDAへのドキュメントのアップロードが禁止されている場合             | ホストからクライアントへのリダイレクト |
| カスタムのURLタイプ         | VDAにカスタムURLタイプ用のアプリケーションがインストールされていない場合  | ホストからクライアントへのリダイレクト |

ホストからクライアントへのリダイレクト機能は、Citrix Receiver for Windows、Receiver for Mac、Receiver for Linux、Receiver for HTML5、Receiver for Chromeでサポートされています。

ホストからクライアントへのリダイレクト機能を使用するには、ユーザーデバイスにWebブラウザ、マルチメディアプレーヤー、またはコンテンツに対応したその他のアプリケーションをインストールする必要があります。ユーザーデバイスがデスクトップアライアンス、シンクライアント、またはゼロクライアントである場合、適切なアプリケーションがインストールされていること、および性能が十分であることを確認してください。

ローカルアプリケーションアクセスを有効にしたユーザーデバイスでは別のコンテンツリダイレクトメカニズムが使用されるため、ホストからクライアントへのリダイレクト機能は必要ありません。

Citrixの各種ポリシーを使用して、不適切なデバイスについてホストからクライアントへのコンテンツリダイレクト機能を禁じます。

ホストからクライアントへのリダイレクト機能は、URLが以下の場合に使用されます。

- アプリケーションにハイパーリンクとして埋め込まれている場合（電子メールメッセージやドキュメントなど）
- VDAアプリケーションのメニューまたはダイアログボックスから選択された場合（アプリケーションでWindows ShellExecuteEx APIが使用されているとき）
- Windowsの [ファイル名を指して実行] ダイアログボックスに入力された場合

Webブラウザ内のURL（WebページのURL内およびWebブラウザのアドレスバーに入力されたURLのどちらでも）に対しては、ホストからクライアントへのリダイレクト機能は使用されません。

## 注意

ユーザーが（[規定のプログラムを設定する]などを使用して）VDA上のデフォルトのWebブラウザを変更した場合、この変更によりアプリケーションのホストからクライアントへのリダイレクト機能が影響を受ける可能性があります。

ホストからクライアントへのリダイレクト機能が有効な場合、URLを開くアプリケーションは、URLの種類とコンテンツの種類の間に関するユーザーデバイスの設定で決まります。次に例を示します。

- コンテンツタイプがHTMLであるHTTP URLは、デフォルトのWebブラウザで開かれます。
- コンテンツタイプがPDFであるHTTP URLは、デフォルトのWebブラウザまたは別のアプリケーションのどちらかで開かれます。

ユーザーデバイスの設定は、ホストからクライアントへのコンテンツリダイレクト機能では制御できません。ユーザーデバイスの設定を制御しない場合は、ホストからクライアントへのコンテンツリダイレクト機能ではなく、Flashリダイレクトおよびマルチメディアリダイレクトを使用してください。

ホストからクライアントへのリダイレクト機能が有効な場合、次の種類のURLはユーザーデバイスでローカルに開かれます。

- HTTP (Hypertext Transfer Protocol)
- HTTPS (Secure Hypertext Transfer Protocol)
- RTSP (RealPlayerおよびQuickTime)
- RTSPU (RealPlayerおよびQuickTime)
- PNM (従来のRealPlayer)
- MMS (Microsoft Media Server)

ホストからクライアントへのリダイレクト機能のURLタイプのリストを変更して、カスタムURLタイプなどのURLタイプを削除および追加できます。

ホストからクライアントへのリダイレクト機能を有効にするには、まずCitrixポリシー設定を有効にします。

ホストからクライアントへのリダイレクトポリシーの設定については、[ファイルリダイレクトのポリシー設定](#)に記載されています。デフォルトでは、無効になっています。

さらに、VDAのOSによっては、サーバーVDAのレジストリキーとグループポリシーの設定も必要になります。

- サーバーVDAがWindows Server 2008 R2 SP1である場合、レジストリキーとグループポリシーを設定する必要はありません。
- サーバーVDAがWindows Server 2012、Windows Server 2012 R2、またはWindows Server 2016である場合、レジストリキーとグループポリシーの設定が必要になります。

## 警告

レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

1. 以下の「Reg file start」と「Reg file end」の間にあるテキストをコピーして、メモ帳に貼り付けます。
2. [名前を付けて保存] で [ファイルの種類] を [すべてのファイル]、[ファイル名] をServerFTA.regに指定して、メモ帳ファイルを保存します。
3. Active Directoryのグループポリシーを使用して、ServerFTA.regファイルをサーバーに配布します。



- Reg file start --

Windows Registry Editor Version 5.00

[HKEY\_CLASSES\_ROOT\ServerFTAHTML\shell\open\command]

@="\"C:\\Program Files (x86)\\Citrix\\system32\\iexplore.exe\" %1"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\ServerFTA]

@="ServerFTA"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\ServerFTA\Capabilities]

"ApplicationDescription"="Server FTA URL."

"ApplicationIcon"="C:\\Program Files (x86)\\Citrix\\system32\\iexplore.exe,0"

"ApplicationName"="ServerFTA"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\ServerFTA\Capabilities\URLAssociations]

"http"="ServerFTAHTML"

"https"="ServerFTAHTML"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\RegisteredApplications]

"Citrix.ServerFTA"="SOFTWARE\Citrix\ServerFTA\Capabilities"

-- Reg file end --

XMLファイルを作成します。このXMLファイルに、以下の「xml file start」と「xml file end」の間にあるテキストをコピーして貼り付けて、ServerFTAdefaultPolicy.xmlという名前で保存します。



```

-- xml file start --

<?xml version="1.0" encoding="UTF-8"?>

<DefaultAssociations>

<Association Identifier="http" ProgId="ServerFTAHTML" ApplicationName="ServerFTA" />

<Association Identifier="https" ProgId="ServerFTAHTML" ApplicationName="ServerFTA" />

</DefaultAssociations>

-- xml file end --

```

現在のグループポリシー管理コンソールで、[コンピューターの構成] > [管理用テンプレート] > [Windowsコンポーネント] > [エクスプローラー] > [既定の関連付け構成ファイルの設定] の順に開いて、作成したServerFTAdefaultPolicy.xmlファイルを指定します。

ホストからクライアントへのリダイレクト機能のURLタイプのリストを変更するには、サーバーVDAで次のレジストリキーを設定します。

キー：HKLM\Software\Wow6432Node\Citrix\SFTA

リストからURLタイプを削除するには、DisableServerFTAとNoRedirectClassesを設定します。

値の名前：DisableServerFTA  
種類：REG\_DWORD  
値のデータ：1

値の名前：NoRedirectClasses  
種類：REG\_MULTI\_SZ  
データ：http、https、rtsp、rtspu、pnm、mmsのいずれかの組み合わせを指定します。1つの行に1つの値を入力してください。次に例を示します。

http

https

rtsp

リストにURLタイプを追加するには、ExtraURLProtocolsを設定します。

値の名前：ExtraURLProtocols  
種類：REG\_MULTI\_SZ

データ：URLタイプの組み合わせを指定します。各URLタイプにはサフィックスとして「://」を追加し、複数の値はセミコロンで区切って入力します。次に例を示します。

```
customtype1://;customtype2://
```

特定のWebサイトのセットについてホストからクライアントへのリダイレクト機能を有効にするには、サーバーVDAで次のレジストリキーを設定します。

キー：HKLM\Software\Wow6432Node\Citrix\SFTA

値の名前：ValidSites

種類：REG\_MULTI\_SZ

データ：FQDN（完全修飾ドメイン名：Fully-Qualified Domain Name）の組み合わせを指定します。1つの行に1つのFQDNを入力してください。FQDNには、左端にのみワイルドカードを含めることができます。ワイルドカードを含むFQDNは単一レベルのドメインと照合されます。これはRFC 6125の規則に準拠しています。次に例を示します。

www.example.com

\*.example.com



# WindowsデスクトップOSのためのGPUアクセラレーション

Jan 04, 2017

するX 3D Proを使用することで、グラフィックアプリケーションを仮想デスクトップ上で提供したりデスクトップOSマシン上のアプリケーションとして配信したりできます。HDX 3D Proは、デスクトップ、ブレード、およびラックワークステーションを含む物理ホストコンピューターと、GPUパススルー機能が有効なXenServer仮想マシン、および仮想GPUを持つXenServer仮想マシンをサポートします。

XenServerのGPUパススルー機能を使用すると、グラフィック処理ハードウェアに排他的にアクセスする仮想マシンを作成できます。ハイパーバイザーに複数のGPUを装着して、各仮想マシンにGPUを1つずつ割り当てることができます。

XenServerのvGPU機能を使用すると、複数の仮想マシンで単一の物理GPUによるグラフィック処理能力に直接アクセスできるようになります。このハードウェアGPU共有により、より専門的で複雑な設計作業を行うユーザーに最適なWindows 7またはWindows Server 2008 R2 SP1デスクトップが提供されます。GPU共有では、NVIDIA GRID K1およびK2のカードがサポートされ、非仮想化オペレーティングシステムで動作するものと同じNVIDIAグラフィックドライバーが使用されます。

HDX 3D Proの機能は以下のとおりです。

- WANおよびワイヤレス接続でのパフォーマンスを最適化するAdaptive H.264ベースの深圧縮。HDX 3D Proのデフォルトでは、CPUベースの深圧縮が使用されます。これにより、ネットワークの状態に応じて最適な圧縮が動的に適用されます。H.264ベースの深圧縮コーデックでは、NVIDIA GPU上のCUDAコアによるレンダリングが使用されません。深圧縮コーデックはCPU上で実行され、帯域幅消費が効率化されます。

- 特殊なアプリケーションのための無損失圧縮オプション。HDX 3D ProではCPUベースの無損失コーデックも提供され、医療用画像処理などピクセル単位での精密なグラフィックが求められるアプリケーションがサポートされます。無損失圧縮はネットワークおよび処理リソースに対する負荷が非常に高いため、特殊なアプリケーションでのみ使用することをお勧めします。

無損失圧縮を使用すると、以下のように動作します。

- 表示しているフレームに非可逆圧縮が適用されているのか無損失圧縮が適用されているのかを示すインジケータがユーザーの通知領域に表示されます。このインジケータは、ポリシーの[表示品質]設定で[操作時は低品質]が選択されている場合に便利です。送信されたフレームが無損失の場合、このインジケータが緑色になります。
- ユーザーは、無損失スイッチを使ってセッション内でいつでも[常に無損失]モードを有効にできます。セッション中に無損失圧縮の有効/無効を切り替えるには、通知領域のインジケータを右クリックするか、Alt+Shift+1キーを押します。

無損失圧縮の場合：ポリシーで指定されているコーデックに関係なく、無損失コーデックが使用されます。

非可逆圧縮の場合：デフォルトのコーデックまたはポリシーで指定されているコーデックが使用されます。

無損失スイッチの設定は保持されず、次のセッションではリセットされます。すべての接続で無損失コーデックが使用されるようにするには、ポリシーの[表示品質]設定で[常に無損失]を選択します。

- [7.6 FP3] で、デフォルトのショートカットであるALT+SHIFT+1を無効にし、セッション内で無損失を選択または選択解除できます。

HKLM\SOFTWARE\Citrix\HDX3D\LLIndicatorで新しいレジストリ設定を構成します。

- 名前：HKLM\_HotKey、種類：String
- ショートカットの組み合わせを構成する形式は、C=0|1, A=0|1, S=0|1, W=0|1, K=valです。キーはコンマ「,」で区切る必要があります。キーの順番は関係ありません。
- A、C、S、W、およびKはキーです。ここで、C=Control、A=ALT、S=SHIFT、W=Win、およびK=aが有効なキーです。Kに対して使用できる値は、0~9、a~z、およびすべての仮想キーコードです。仮想キーコードについて詳しくは、MSDNのVirtual-Key Codesを参照してください。
- 次に例を示します。
  - F10に、K=0x79を設定
  - Ctrl + F10に、C=1, K=0x79を設定
  - Alt + Aに、A=1, K=aまたはA=1, K=AまたはK=A, A=1を設定
  - Ctrl + Alt + 5に、C=1, A=1, K=5またはA=1, K=5, C=1を設定
  - Ctrl + Shift + F5, A=1, S=1, K=0x74を設定

**注意：**レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

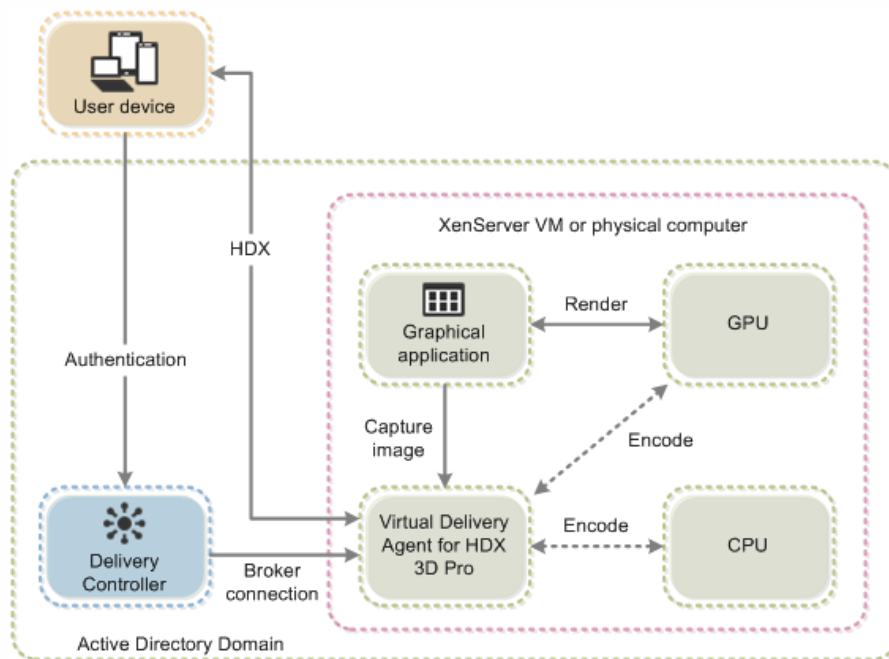
- 複数および高解像度モニターのサポート。Windows 7およびWindows 8のデスクトップでは、最大で4つのモニターが構成されたユーザーデバイスがサポートされます。ユーザーはそれらのモニターを自由に配置でき、解像度や向きが異なるモニターを組み合わせで使用できます。モニターの数は、ホストコンピューターのGPU、ユーザーデバイス、および使用できる帯域幅による制限を受けます。HDX 3D Proでは、ホストコンピューター上のGPUでサポートされるすべてのモニター解像度がサポートされます。

また、Windows XPデスクトップでは、デュアルモニター構成が限定的にサポートされます。これについて詳しくは、[Windows XPまたはWindows Vista上のVDA](#)を参照してください。

- 動的解像度 仮想デスクトップまたはアプリケーションのウィンドウのサイズを任意に変更できます。注：解像度は、VDAのセッションウィンドウのサイズを変更することでのみ変更できます。VDAセッション内での解像度の変更（[コントロールパネル] > [デスクトップのカスタマイズ] > [ディスプレイ] > [画面の解像度]で変更）はサポートされていません。
- NVIDIA Keplerアーキテクチャのサポート。GPUパススルーおよびGPU共有では、NVIDIA GRID K1およびK2のカードがサポートされます。NVIDIA GRID vGPUを使用すると、複数の仮想マシンで単一の物理GPUに同時に直接アクセスできます。このとき、仮想化されていないオペレーティングシステムで動作するものと同様NVIDIAグラフィックドライバーが使用されます。
- Virtual Direct Graphics Acceleration (vDGA) を使ったVMware vSphereおよびVMware ESXのサポート - RDSおよびVDIの両方のワークロードで、vDGAがサポートされます。HDX 3D ProでVirtual Shared Graphics Acceleration (vSGA) を使用する場合は、単一モニターのみサポートされます。vSGAを大きな3Dモデルで使用する、APIインターセプト技術の使用によりパフォーマンス上の問題が生じることがあります。詳しくは、「[VMware vSphere 5.1 - Citrix Known Issues](#)」を参照してください。

以下の図を参照してください。

- ホストコンピューターは、Delivery Controllerと同じActive Directoryドメインに属している必要があります。
- ユーザーがCitrix Receiverにログインして仮想アプリケーションまたはデスクトップにアクセスすると、Controllerがユーザーを認証してVDA for HDX 3D Proと通信し、グラフィックアプリケーションをホストしているコンピューターへの接続を仲介します。VDA for HDX 3D Proはホスト上の適切なハードウェアを使って、デスクトップ全体またはグラフィックアプリケーションだけのビューを圧縮します。
- デスクトップまたはアプリケーションのビューおよびそれに対するユーザーの応答は、Citrix ReceiverとVDA for HDX 3D Pro間の直接HDX接続を介して、ホストコンピューターとユーザーデバイス間で転送されます。



VDA for Windows Desktop OSをインストーラーのグラフィカルインターフェイスでインストールする場合は、[HDX 3D Pro] ページで [はい] をクリックします。コマンドラインでインストールする場合は、XenDesktopVdaSetup.exeコマンドに/enable\_hdx\_3d\_proオプションを指定します。

HDX 3D Proをアップグレードするには、既存のHDX 3D for Professional GraphicsコンポーネントとVDAの両方をアンインストールしてからVDA for HDX 3D Proをインストールしてください。同様に、標準のVDA for Windows Desktop OSをVDA for HDX 3D Proに切り替える場合も、標準のVDAをアンインストールしてからVDA for HDX 3D Proをインストールしてください。

NVIDIA GRID APIでは、GPUのフレームバッファに対する直接アクセスが提供され、スムーズでインタラクティブなユーザーエクスペリエンスのための最速のフレームレートが提供されます。NVIDIAドライバーをインストールしてからVDA for HDX 3D Proをインストールすると、NVIDIA GRIDがデフォルトで有効になります。

仮想マシン上でNVIDIA GRIDを有効にするには、デバイスマネージャーでMicrosoft基本ディスプレイアダプターを無効にします。Montereyenable.exe -enable -noreset コマンドを実行してからVDAを再起動します。

VDA for HDX 3D ProをインストールしてからNVIDIAドライバーをインストールすると、NVIDIA GRIDが無効になります。この場合は、NVIDIA社から提供されるMontereyenableツールを使ってNVIDIA GRIDを有効にします。

NVIDIA GRIDを無効にするには、Montereyenable.exe -disable -noreset コマンドを実行してからVDAを再起動します。

マルチモニター環境でHDX 3D Proを使用するには、ユーザーデバイスに接続されているモニター数以上のモニターがホストコンピューター側に構成されている必要があります。ホストコンピューター側で構成されているモニターは、物理モニターまたは仮想モニターのどちらでも構いません。

ユーザーがグラフィックアプリケーションの仮想デスクトップまたはアプリケーションに接続している間は、ホストコンピューターにモニター（物理または仮想のいずれも）を接続しないでください。これを行うと、ユーザーのセッションが不穩定になることがあります。

グラフィックアプリケーションセッションを実行しているときにデスクトップの解像度を変更しないようにユーザーに通知してください。アプリケーションセッションを閉じた後、[Citrix Receiver - Desktop Viewer基本設定] ダイアログボックスでDesktop Viewerウィンドウの解像度を変更できます。

支店など、帯域幅が制限された接続を複数のユーザーで共有している場合、ポリシーの[セッション全体の最大帯域幅]設定を使用して、各ユーザーが使用できる帯域幅を制限することをお勧めします。これにより、ユーザーがログオンしたりログオフしたりするときに、使用可能な帯域幅が大きく変動しなくなります。HDX 3D Proでは使用可能なすべての帯域幅が使用されるため、ユーザーのセッション中に使用可能な帯域幅が大きく増減するとパフォーマンスが低下します。

たとえば、60Mbpsの接続を20人のユーザーで共有する場合、各ユーザーが使用できる帯域幅は、同時接続ユーザーの数に応じて3Mbps~60Mbpsの間で変動します。この場合におけるユーザーエクスペリエンスを最適化するには、各ユーザーがピーク時に必要とする帯域幅を調べて、常時この値でユーザーを制限します。

ユーザーが3Dマウスを使用する場合は、汎用USBリダイレクト仮想チャンネルの優先度を0にすることをお勧めします。仮想チャンネルの優先度を変更する方法については、[CTX128190](#)を参照してください。

# WindowsサーバーOSのためのGPUアクセラレーション

Jan 04, 2017

HDX 3D Pro機能により、WindowsサーバーOSのセッションで実行しているグラフィック処理アプリケーションで、サーバー上のGPU (Graphics Processing Unit) リソースを使用できるようになります。OpenGL、DirectX、Direct3D、およびWindows Presentation Foundation (WPF) の処理をサーバーのGPUに移すことで、グラフィック処理によりサーバーのCPU速度が低下することを回避できます。また、ワークロードがCPUとGPUで分担されるため、サーバーでより多くのグラフィック処理が可能になります。

HDX 3D Proにより、複数のユーザーがグラフィックカードを共有できるようになります。HDX 3D ProとXenServerのGPUパススルー機能を使用すると、単一のサーバーで複数のグラフィックカード (各仮想マシンに1つずつ) をホストできるようになります。

このトピックの説明にはレジストリの編集が含まれています。レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があります。Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

GPU共有により、リモートデスクトップセッションで動作するOpenGLアプリケーションおよびDirectXアプリケーションでGPUハードウェアによるレンダリング処理が可能になります。これにより、以下の効果があります。

- ベアメタルまたは仮想マシン上で使用でき、アプリケーションのスケラビリティとパフォーマンスが向上します。
- 複数の同時接続セッションでGPUリソースを共有できます (ほとんどのユーザーは専用GPUのレンダリングパフォーマンスを必要としません)。
- 特別な設定は必要ありません。

複数のGPUを持つグラフィックカードを装着したり、複数のグラフィックカードを装着したりして、ハイパーバイザー上に複数のGPUをインストールして各GPUを特定の仮想マシンに (1対1で) 割り当てることができます。ただし、サーバー上で異なるグラフィックカードを混在させることは推奨されません。

仮想マシンでは、GPUへの直接パススルーアクセスが必要です。この機能は、Citrix XenServerまたはVMware vSphereで提供されます。HDX 3D ProとGPUパススルーを併用すると、サーバー上の各GPUで単一のマルチユーザー仮想マシンをサポートできます。

GPU共有は、特定のグラフィックカードに依存するものではありません。

- ハイパーバイザー上では、そのハイパーバイザーのGPUパススルー機能でサポートされるハードウェアプラットフォームとグラフィックカードを選択してください。XenServerのGPUパススルー機能でテストされたハードウェアの一覧については、「[GPU Pass-through Devices](#)」を参照してください。
- When running on bare metal, it is recommended to have a single display adapter enabled by the operating system. If multiple GPUs are installed on the hardware, disable all but one of them using Device Manager.

GPU共有でのスケラビリティは、以下の要素により異なります。

- 実行するアプリケーション
- 消費されるビデオRAMの量
- グラフィックカードの処理能力

たとえば、NVIDIA Q6000およびM2070Qカードを使用してESRI ArcGISなどのアプリケーションを実行する場合、8~10人のユーザーがサポートされました。これらのカードでは、6GBのビデオRAMが提供されます。より新しいNVIDIA GRIDのビテ

オRAMは8GBで、処理能力も向上しています（より多くのCUDAコアで構成されている）。NVIDIA GRID K2カードの場合、GRID K2カードあたり最大20人のユーザーで良好なパフォーマンスが得られました。ほかのアプリケーションを使用する場合のスケラビリティはさらに高く、上位クラスのGPUで32人の同時ユーザーがサポートされます。

一部のアプリケーションでは、ビデオRAMの不足をより効果的に処理できます。ハードウェアの負荷が過剰になると、グラフィックカードドライバーが不安定になったり異常停止したりすることがあります。このような問題を避けるには、同時接続ユーザーの数を制限してください。

GPUアクセラレーションが正しく動作しているかどうかを確認するには、GPU-Zなどのサードパーティ製ツールを使用できます。このツールは、<http://www.techpowerup.com/gpuz/>で提供されています。

DirectX、Direct3D、およびWPFレンダリングは、DDI (Display Driver Interface) Version 9ex、10、または11をサポートするGPUが搭載されたサーバーでのみ使用可能です。

- Windows Server 2008 R2では、DirectXおよびDirect3Dで単一GPUを使用するために特別な設定は不要です。
- Windows Server 2012のRD Session Hostサーバー上のリモートデスクトップサービス (RDS) セッションでは、デフォルトのアダプターとしてMicrosoft基本レンダリングドライバーが使用されます。Windows Server 2012上のRDSセッションでGPUを使用するには、グループポリシーの [ローカルコンピューターポリシー] > [コンピューターの構成] > [管理用テンプレート] > [Windowsコンポーネント] > [リモートデスクトップサービス] > [リモートデスクトップセッションホスト] > [リモートセッション環境] で [Use the hardware default graphics adapter for all Remote Desktop Services sessions] を有効にします。
- WPFアプリケーションでのレンダリングにサーバーのGPUが使用されるようにするには、WindowsサーバーOSセッションを実行するサーバー上で以下のレジストリキーを設定します。
  - [HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\CtxHook\Applnit\_Dlls\Multiple Monitor Hook]  
"EnableWPFHook"=dword:00000001
  - [HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\CtxHook\Applnit\_Dlls\Multiple Monitor Hook]  
"EnableWPFHook"=dword:00000001

このリリースでは、ユーザーセッション内で動作するCUDAおよびOpenCLアプリケーション用のGPUアクセラレーションが試験的に提供されます。この機能はデフォルトで無効になっていますが、テストまたは評価のために有効にすることができず。

CUDAアクセラレーション機能を有効にするには、以下のレジストリを設定します。

- [HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\CtxHook\Applnit\_Dlls\Graphics Helper] "CUDA"=dword:00000001
- [HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\CtxHook\Applnit\_Dlls\Graphics Helper]  
"CUDA"=dword:00000001

OpenCLアクセラレーション機能を有効にするには、以下のレジストリを設定します。

- [HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\CtxHook\Applnit\_Dlls\Graphics Helper] "OpenCL"=dword:00000001
- [HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\CtxHook\Applnit\_Dlls\Graphics Helper]  
"OpenCL"=dword:00000001

# OpenGLソフトウェアアクセラレータ

Jan 04, 2017

OpenGLソフトウェアアクセラレータは、ArcGIS、Google Earth、Nehe、Maya、Blender、Voxler、CAD、およびCAMなどのOpenGLアプリケーションで使用するソフトウェアラスタライザーです。OpenGLソフトウェアアクセラレータを使用すると、グラフィックカードを装着しなくてもユーザーに良好なユーザーエクスペリエンスを提供できる場合があります。

**重要：**OpenGLソフトウェアアクセラレータは現状有姿のまま提供されます。このため、すべてのアプリケーションで動作確認を行ってください。このソフトウェアは、一部のアプリケーションで動作しない場合があります。この機能は、Windows OpenGLラスタライザーで十分なパフォーマンスが得られない場合の解決策として使用してください。OpenGLソフトウェアアクセラレータがアプリケーションで動作する場合は、GPUハードウェアのコストを削減する手段として使用できます。

OpenGLソフトウェアアクセラレータはインストールメディアのSupportフォルダーに収録されており、すべてのVDAプラットフォームをサポートしています。

OpenGLソフトウェアアクセラレータは、以下の場合に使用します。

- XenServerやほかのハイパーバイザーの仮想マシン上で動作するOpenGLアプリケーションで十分なパフォーマンスが得られない場合は、OpenGLアクセラレータを使用してテストを行います。一部のアプリケーションでは、Windowsに付属のMicrosoft OpenGLソフトウェアよりも高いパフォーマンスが得られる場合があります。これは、OpenGLアクセラレータでSSE4.1およびAVXが使用されるためです。OpenGLアクセラレータは、Version 2.1までのOpenGLを使用するアプリケーションをサポートします。
- ワークステーション上で動作するアプリケーションでは、まずそのワークステーションのグラフィックアダプターで提供されるデフォルトのOpenGLサポート機能を使用してテストを行います。一般的に、最新バージョンのグラフィックカードを使用すると最高のパフォーマンスが得られます。グラフィックカードのバージョンが低い場合、または十分なパフォーマンスが得られない場合は、OpenGLソフトウェアアクセラレータを使用してテストを行います。
- 3D OpenGLアプリケーションがCPUベースのソフトウェアラスタライザーで適切に配信されていない場合は、OpenGLのGPUハードウェアアクセラレーションを使用することでパフォーマンスが向上することがあります。この機能は、ベアメタルまたは仮想マシン上で使用できます。

# オーディオ機能

Jan 04, 2017

ポリシーに以下の設定項目を追加して、HDXのオーディオ機能を最適化できます。これらの設定項目の使用方法、およびほかのポリシー設定項目との依存関係について詳しくは、「[オーディオのポリシー設定](#)」、「[帯域幅のポリシー設定](#)」および「[マルチストリーム接続のポリシー設定](#)」を参照してください。

**重要:** 多くのオーディオ機能はICAストリームを使って転送され、ほかのICAトラフィックと同じ方法で保護されます。NetScaler Access GatewayがIPス上にない場合、ユーザーデータグラムプロトコル (UDP) オーディオでは、これとは別の保護されていない転送メカニズムが使用されます。NetScaler Access GatewayがXenAppおよびXenDesktopのリソースにアクセスするよう構成されている場合、エンドポイントデバイスとNetScaler Access Gateway間のオーディオトラフィックはDTLSプロトコルで保護されます。

一般的に、音質を高くするほど、オーディオデータの転送に必要な帯域幅が大きくなり、サーバーのCPUにも負担がかかります。オーディオデータを圧縮すると、セッションのパフォーマンスと音質とのバランスを考慮しながら、ユーザーの操作感を最適化できます。これを行うには、サウンドファイルに適用する圧縮レベルを制御するには、Citrixポリシーを使用します。

デフォルトでは、[音質] 設定で [高 - 高品位オーディオ] が指定されています。これにより、HiFiステレオオーディオが提供されますが、ほかの設定よりも多くの帯域幅が消費されます。最適化されていないボイスチャットアプリケーションやビデオチャットアプリケーション (ソフトフォンなど) でこの設定を使用すると、リアルタイムの音声通信に不適切な遅延が発生することがあります。

たとえば、ダイヤルアップで接続するユーザーグループとLANまたはWANで接続するユーザーグループに、異なるポリシーを作成します。ダイヤルアップ接続では帯域幅が限られているため、低い音質レベルを指定したポリシーを適用して、音質よりもセッションパフォーマンスを向上させます。LANまたはWANで接続するユーザーグループに対しては、高い音質レベルを指定したポリシーを適用します。

設定について詳しくは、「[オーディオのポリシー設定](#)」を参照してください。ユーザーデバイス側のクライアントオーディオ設定が有効になっていることを確認してください。詳しくは、「[ユーザーデバイス側のオーディオ設定ポリシー](#)」を参照してください。

サーバー上で実行しているアプリケーションからユーザーデバイス上のスピーカーまたはヘッドフォンなどのサウンドデバイスでオーディオが再生されるようにするには、ポリシーに [クライアントオーディオリダイレクト] 設定を追加して [許可] (デフォルト) を指定します。

クライアントオーディオマッピングによりサーバーおよびネットワークの負荷が増大しますが、[クライアントオーディオリダイレクト] 設定で [禁止] を指定するとHDXのすべてのオーディオ機能が無効になります。

設定について詳しくは、「[オーディオのポリシー設定](#)」を参照してください。ユーザーデバイス側のクライアントオーディオ設定が有効になっていることを確認してください。詳しくは、「[ユーザーデバイス側のオーディオ設定ポリシー](#)」を参照してください。

ユーザーデバイス上のマイクなどのサウンド入力デバイスを使って録音できるようにするには、ポリシーに [クライアントマイクリダイレクト] 設定を追加して [許可] (デフォルト) を指定します。

ユーザーデバイスとの信頼関係が設定されていないサーバー上のセッションでマイクを使用しようとする、セキュリティに関する警告がユーザーに表示されます。この場合、クライアント側のマイクへのアクセスを許可するかどうかをユーザーが選択できます。この警告は、ユーザーがCitrix Receiver側で無効にできます。

設定について詳しくは、「[オーディオのポリシー設定](#)」を参照してください。ユーザーデバイス側のクライアントオーディオ設定が有効になっていることを確認してください。詳しくは、「[ユーザーデバイス側のオーディオ設定ポリシー](#)」を参照してください。

ポリシーの [オーディオプラグアンドプレイ] 設定では、録音やサウンド再生のための複数のオーディオデバイスの使用を許可または禁止します。この設定項目は、デフォルトで [有効] になっています。

この設定項目は、WindowsサーバーOSマシンのみ適用されます。

設定について詳しくは、「[オーディオのポリシー設定](#)」を参照してください。

ポリシーの [オーディオリダイレクトの最大帯域幅 (Kbps)] 設定では、クライアント側デバイスによるオーディオの再生や録音で使用可能な最大帯域幅を、キロビット/秒 (Kbps) 単位で指定します。ポリシーの [オーディオリダイレクトの最大帯域幅 (%) ] 設定では、クライアント側デバイスによるオーディオの再生や録音で使用可能な最大帯域幅を、セッション全体に対する割合で指定します。これらの設定項目には、デフォルトで0が指定されており、帯域幅に制限はありません。両方の設定項目を構成した場合、より高い制限 (より小さい値) の設定が適用されます。

設定について詳しくは、「[帯域幅のポリシー設定](#)」を参照してください。ユーザーデバイス側のクライアントオーディオ設定が有効になっていることを確認してください。詳しくは、「[ユーザーデバイス側のオーディオ設定ポリシー](#)」を参照してください。

ポリシーの [UDPでのオーディリアルタイムトランスポート] 設定は、デフォルトで [有効] が選択されています (インストール時に選択した場合)。これにより、サーバーのUDPポートが開き、[UDPでのオーディリアルタイムトランスポート] 設定が有効な接続でそのポートが使用されます。ネットワークで輻輳やパケット損失などが生じる環境で最適なユーザーエクスペリエンスを提供するため、オーディオのUDP/RTPを構成することをお勧めします。

**重要 :** NetScaler Access Gatewayがパス上にない場合、UDPで転送されるオーディオデータは暗号化されません。NetScaler Access GatewayがXenAppおよびXenDesktopのリソースにアクセスするよう構成されている場合、エンドポイントデバイスとNetScaler Access Gateway間のオーディオトラフィックはDTLSプロトコルで保護されます。

ポリシーの [オーディオUDPポートの範囲] 設定では、Virtual Delivery Agent (VDA) でユーザーデバイスとのオーディオパケットデータの送受信に使用されるポート番号の範囲を指定します。

デフォルトでは、16500~16509の範囲が指定されています。

[UDPでのオーディリアルタイムトランスポート] 設定については「[オーディオのポリシー設定](#)」を、[オーディオUDPポートの範囲] 設定については「[マルチストリーム接続のポリシー設定](#)」を参照してください。ユーザーデバイス側の [Client audio settings] が有効になっていることを確認してください。詳しくは、「[ユーザーデバイス側のオーディオ設定ポリシー](#)」を参照してください。

1. 「[グループポリシーオブジェクトテンプレートによるReceiverの構成](#)」の手順に従って、グループポリシーテンプレートを



ロードします。

2. グループポリシーエディターで、 [管理用テンプレート] > [Citrix Components] > [Citrix Receiver] > [User Experience] の順に開きます。
3. [Client audio settings] を開き、 [未構成]、 [有効]、または [無効] をクリックします。
  - **設定しない**： デフォルトでは、オーディオリダイレクトは高品質オーディオ、または以前に構成したカスタムのオーディオ設定で有効になります。
  - **Enabled**。 オーディオリダイレクトは、選択したオプションで有効になります。
  - **無効**。 オーディオリダイレクトは無効化されます。
4. [有効] をクリックした場合は、音質を選択します。 UDPオーディオでは、 [中] (デフォルト) を使用してください。
5. UDPオーディオの場合のみ、 [リアルタイムトランスポートを有効にする] チェックボックスをオンにして、ローカルの Windows ファイアウォールを通過するための着信ポートの範囲を指定します。
6. NetScaler Access GatewayでUDPオーディオを使用するには、 [ゲートウェイ経由でのリアルタイムトランスポートを許可する] チェックボックスをオンにします。 NetScaler Access GatewayはDTLSで構成する必要があります。 詳しくは、 「[UDP Audio Through a Netscaler Gateway](#)」 を参照してください。

エンドポイントデバイスで上記の変更を行う制御権を持っていない場合 (BYODの場合や家庭用コンピューターの場合など)、管理者としてStoreFrontのdefault.ica属性を使用してUDPオーディオを有効にします。

1. StoreFront マシンで、メモ帳などのエディターを使用してC:\inetpub\wwwroot\Citrix\<ストア名>\App\_Data\default.icaを開きます。
2. [Application] セクションに以下の項目を記入します。

```
; This is to enable Real-Time Transport
```

```
EnableRtpAudio=true
```

```
; This is to Allow Real-Time Transport Through gateway
```

```
EnableUDPTThroughGateway=true
```

```
; This is to set audio quality to Medium
```

```
AudioBandwidthLimit=1-
```

```
; UDP Port range
```

```
RtpAudioLowestPort=16500
```

```
RtpAudioHighestPort=16509
```

UDPオーディオは、default.icaの編集で有効になっている場合、そのストアを使用するすべてのユーザーに対して有効化されます。

オーディオまたはビデオ会議にユーザーが参加したときに、音声にエコーがかかって聞こえることがあります。通常、この問題はスピーカーとマイクが近すぎる場合に発生します。このため、オーディオまたはビデオ会議ではヘッドセットを使用することをお勧めします。

HDXには、会議中のエコーを最小限に抑えるためのエコーキャンセル機能が用意されており、デフォルトで有効になっています。エコーキャンセル機能の効果は、スピーカーとマイクとの距離により異なります。スピーカーとマイクは、近づけすぎず遠ざけすぎず、最適に配置してください。

エコーキャンセル機能を無効にするには、レジストリ設定を変更します。レジストリを編集する場合は細心の注意が必要です。レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

1. ユーザーデバイス上でレジストリエディターを開き、以下のレジストリキーを選択します。

- 32ビットシステム : HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\ICA  
Client\Engine\Configuration\Advanced\Modules\ClientAudio\EchoCancellation
- 64ビットシステム : HKEY\_LOCAL\_MACHINE\SOFTWARE Wow6432Node\Citrix\ICA  
Client\Engine\Configuration\Advanced\Modules\ClientAudio\EchoCancellation

2. [値のデータ] ボックスの値をFALSEに変更します。

# ネットワークトラフィックの優先度

Jan 04, 2017

QoS (Quality of Service) 機能をサポートするルーターを使って単一セッションに複数の接続を使用する場合、ネットワークトラフィックの優先度を割り当てることができます。ユーザーデバイスとサーバー間のICAトラフィックでは、4つのTCP/IPストリーム（リアルタイム、インタラクティブ、バックグラウンド、およびバルク）と1つのUDP/RTPストリーム（ボイス）を使用できます。各仮想チャネルには特定の優先度が割り当てられており、対応する接続を使って転送が行われます。これらの仮想チャネルには、使用されるTCPポート番号に基づいて個別に優先度を設定できます。

Windows 8およびWindows 7マシンにインストールしたVirtual Delivery Agent (VDA) では、複数チャネルのストリーム接続がサポートされます。ネットワーク管理者に問い合わせ、[マルチポートポリシー] 設定で指定したCGP (Common Gateway Protocol) ポートが、ネットワークルーター上で正しく割り当てられていることを確認してください。

QoS (Quality of Service) は、セッション画面の保持機能のポートまたはCGPポートが複数設定されている環境でのみサポートされます。

注意：この機能を使用する場合は、IPsec (Internet Protocol Security) またはSSL (Secure Sockets Layer) を使用することをお勧めします。SSL接続がサポートされるのは、マルチストリームをサポートするNetScaler Gatewayを通過するトラフィックのみです。企業内ネットワークでは、SSLを使用したマルチストリーム接続はサポートされません。マルチストリーム接続のQoSを設定するには、ポリシーに以下のCitrixポリシー設定を追加します（詳しくは、[マルチストリーム接続のポリシー設定](#)を参照してください）。

- マルチポートポリシー- 複数接続を介したICAトラフィックで使用されるポートおよびそのネットワーク優先度を指定します。
  - [CGPデフォルトポートの優先度] ボックスの一覧で優先度を選択します。デフォルトでは、プライマリポート (2598) に優先度 [高] が設定されます。
  - [CGPポート1]、[CGPポート2]、および [CGPポート3] ボックスに追加のCGPポートを入力して、それぞれ優先度を選択します。各ポートには異なる優先度を設定する必要があります。

VDA側のファイアウォールで、追加したTCPトラフィックを明示的に許可する必要があります。

- マルチストリームコンピューター設定- この設定は、デフォルトでは無効になっています。Citrix CloudBridgeでマルチストリーム機能をサポートする場合は、この設定を使用する必要はありません。このポリシー設定は、サードパーティ製のルーターや従来のBranch Repeaterを使用する環境でQoS (Quality of Service) 優先度を指定するときに使用できます。
- マルチストリームユーザー設定 — この設定は、デフォルトでは無効になっています。

ポリシーの設定を反映させるには、ユーザーがネットワークに再ログインする必要があります。

# USBとクライアント側ドライブの考慮事項

Jan 04, 2017

HDX USBデバイスリダイレクト機能を使用すると、デスクトップでホストされるアプリケーションや仮想デスクトップを使用しているときに、ローカルコンピューターに装着したUSBフラッシュドライブにアクセスできるようになります。セッション内で、デジタルカメラなどの画像転送プロトコル (PTP) デバイス、デジタルオーディオプレーヤーやポータブルメディアプレーヤーなどのメディア転送プロトコル (MTP) デバイス、およびPOS (Point-Of-Sale) デバイスを接続して使用できます。

デスクトップでホストされるアプリケーションのセッションでは、ダブルホップUSBはサポートされません。

USBデバイスリダイレクト機能は、Receiver for WindowsまたはReceiver for Linuxを使用しているユーザーに対してのみ有効です。

USBリダイレクトのデフォルトでは、特定のクラスのUSBデバイスでのみ許可され、ほかのクラスのデバイスはリダイレクトされません。詳しくは、Receiverのドキュメントを参照してください。管理者は、リダイレクトするUSBデバイスの一覧を変更して、仮想デスクトップで使用可能になるUSBデバイスの種類を制限できます。

## Important

ユーザーデバイスとサーバー間でセキュリティ境界による分離が不要な環境では、使用を禁止するUSBデバイスの種類についてユーザーに通知してください。

一般的なUSBデバイスリダイレクトするための仮想チャンネルが最適化されており、WAN接続でも良好なパフォーマンスが提供されます。サポートされる機能は、ユーザーデバイス上にインストールされたReceiverにより異なります。最適化された仮想チャンネルは、通常は最適のオプションで、特に高レイテンシ環境には最適です。

SMARTのUSBリダイレクトでSMARTボードを使用する場合、マウスとして処理されます。

ただし、最適化された仮想チャンネルでUSB 3.0デバイスおよびUSB 3.0ポートを使用して、CDM仮想チャンネルを使用してカメラ内のファイルを表示したり、ヘッドセットでオーディオを再生したりできます。USB 3.0デバイスをUSB 2.0ポートに接続した場合も、汎用USBリダイレクトがサポートされます。

最適化された仮想チャンネルを使用できない特殊なデバイスは、汎用USB仮想チャンネルを使用してリダイレクトされます。XenDesktopでテストされているUSBデバイスについては、[CTX123569](#)を参照してください。

Webカメラのヒューマンインターフェイスデバイス (HID) ボタンなど、一部のデバイス固有の機能は、最適化された仮想チャンネルで正しく動作しない場合があります。問題が発生する場合は、汎用USB仮想チャンネルを使用してください。

一部のデバイスはデフォルトではリダイレクトされず、ローカルセッションでのみ使用可能になります。たとえば、ユーザーデバイスの内部USBでシステムボードに装着されたネットワークインターフェイスカードは、リモートで動作する仮想デスクトップでの使用には適しません。

以下のCitrixポリシー設定を使ってUSBサポートを制御します。

- クライアントUSBデバイス最適化規則。GPMCを使用する7.6 FP3で使用可能。たとえば署名デバイスや描画タブレットなど、class=03の入力デバイスで最適化モードはサポートされます。規則が指定されていない場合は、デバイスはインタラクティブモード (02) で制御されます。キャプチャモード (04) は署名デバイスに対する推奨モードです。
- クライアントUSBデバイスリダイレクト。デフォルトは [禁止] です。
- クライアントUSBデバイスリダイレクト規則。規則は汎用USBリダイレクトを使用するデバイスにのみ適用されます。このため、クライアント側ドライブマッピング (CDM) など、専用のリダイレクト機能や最適化されたリダイレクト機能を使用するデバイスには、規則が適用されません。
- クライアントUSBプラグアンドプレイデバイスリダイレクト。デフォルトで [許可] が設定されており、ユーザーセッションでのPTP、MTP、およびPOSデバイスのプラグアンドプレイ機能が有効になります。
- USBデバイスリダイレクトの最大帯域幅 (Kbps)。デフォルトは0 (無制限) です。
- USBデバイスリダイレクトの最大帯域幅 (%)。デフォルトは0 (無制限) です。

汎用USBリダイレクトは、最適化された仮想チャンネルがないUSBデバイスに特化しています。この機能では、任意のUSBデバイスをクライアントマシンから仮想デスクトップにリダイレクトします。これにより、ユーザーはさまざまな種類の汎用USBデバイスを、デスクトップセッションに物理的に接続されているかのように使用できます。

以下の考慮事項があります。

- ユーザーデバイスにデバイスドライバーをインストールする必要はありません
- USBクライアントドライバーはVDAマシン上にインストールされます

この機能はVDA for Desktop OS 7.6のデスクトップセッションでサポートされます。

この機能はまた、次の制限によりVDA for Server OS 7.6のデスクトップセッションでもサポートされます。

- VDAマシンはWindows Server 2012 R2で実行する必要があります
- シングルホップのシナリオだけがサポートされます
- USBクライアントドライバーはRDSH for Windows 2012 R2と互換性がある必要があります
- USBストレージデバイス、オーディオデバイス、スマートカードリーダー、および完全には仮想化されていないデバイスではサポートされません

汎用USBリダイレクトの構成について詳しくは、[CTX137939](#)を参照してください。

1. ポリシーを作成し、[クライアントUSBデバイスリダイレクト]設定を追加して [許可] を選択します。
2. 必要な場合は、[クライアントUSBデバイスリダイレクト規則]設定を追加して、リダイレクトするUSBデバイスの一覧を変更します。
3. ユーザーデバイスにReceiverをインストールするときに、USBサポートを有効にします。前の手順でVirtual Delivery AgentのUSBポリシー規則を指定した場合は、Receiverについて同じ規則を指定します。シンクライアントでのUSBサポートおよびその構成方法については、デバイスの製造元に問い合わせてください。

USBサポート機能が有効になっており、USB関連のユーザー設定でUSBデバイスに自動接続するように設定されている場合は、USBデバイスが自動的にリダイレクトされます。また、デスクトップアプライアンスモードで接続バーが表示されていない場合も、USBデバイスが自動的にリダイレクトされます。ただし、必ずしもすべてのUSBデバイスをリダイレクトするのが適切とは限りません。詳しくは、[CTX124047](#)を参照してください。

ユーザーは、USBデバイスの一覧からデバイスを選択することによって、自動的にリダイレクトされないデバイスを明示的にリダイレクトすることができます。USBデバイスを一覧表示しないようにしたり、USBデバイスをリダイレクトできないようにしたりするには、クライアントおよびVDA上でデバイス規則を定義します。

ReceiverおよびVDAのUSBデバイスリダイレクト規則でリダイレクトするUSBデバイスの種類を指定すると、デフォルトのUSBポリシー規則よりも優先されます。

- ユーザーデバイス側のレジストリを編集します。インストールメディアに収録されている管理テンプレート (ADMファイル、\os\lang\Support\Configuration\icaclient\_usb.adm) により、Active Directoryのグループポリシーを使用してユーザーデバイスを変更できます。
- グループポリシー規則を介して、サーバーOSマシン上のOSの管理者による上書き規則を編集します。グループポリシー管理コンソールは、インストールメディアにあります。
  - x64の場合：DVDルートの\os\lang\x64\Citrix Policy\CitrixGroupPolicyManagement\_x64.msi
  - x86の場合：DVDルートの\os\lang\x86\Citrix Policy\CitrixGroupPolicyManagement\_x86.msi

## 警告

レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があります。Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

製品のデフォルトの規則は、HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\PortICA\GenericUSB\DeviceRulesに格納されています。このデフォルトの規則は変更しないでください。ただし、以下で説明しているように、製品のデフォルトの規則を参照して管理者による上書き規則を作成できます。管理者による上書き規則は、製品のデフォルトの規則よりも先に評されます。

管理者による上書き規則は、HKEY\_LOCAL\_MACHINE\SOFTWARE\Policies\Citrix\PortICA\GenericUSB\DeviceRulesに格納されています。GPOポリシー規則は、{Allow;|Deny;}の後にスペースで区切った一連の=式の形式で設定します。以下のタグがサポートされます。

| タグ       | 説明                                                                                                                   |
|----------|----------------------------------------------------------------------------------------------------------------------|
| VID      | デバイス記述子のベンダーID                                                                                                       |
| PID      | デバイス記述子の製品ID                                                                                                         |
| REL      | デバイス記述子のリリースID                                                                                                       |
| Class    | デバイス記述子またはインターフェイス記述子のクラス。USB Webサイト ( <a href="http://www.usb.org/">http://www.usb.org/</a> ) にあるUSBクラスコードを参照してください |
| SubClass | デバイス記述子またはインターフェイス記述子のサブクラス                                                                                          |
| Prot     | デバイス記述子またはインターフェイス記述子のプロトコル                                                                                          |

新しいポリシー規則を作成する場合、以下の点に注意してください。

- 大文字と小文字は区別されません。
- 規則の末尾に、#で始まる任意のコメントを追加できます。区切り文字は不要で、コメントは無視されます。
- 空白行およびコメントのみの行は無視されます。
- 区切り文字にはスペースが使用されますが、番号または識別子の間には使用できません。たとえば、Deny: Class = 08 SubClass=05」は有効ですが、「Deny: Class=0 Sub Class=05」は無効です。
- タグには等号 (=) を使用する必要があります (VID=1230など)。
- 各規則を1行ずつ記述するか、同一行に記述する場合はセミコロンで区切られたリスト形式である必要があります。

## Important

ADMテンプレートを使用する場合は、規則を単一行に (セミコロン区切りのリストとして) 作成する必要があります。

マストレージデバイスなど、最適化されたデバイスを使用する場合は、一般的にポリシーの規則を使用せず、専用のCDMチャンネルを使用してデバイスをリダイレクトします。ただし、これらについては次のいずれか方法で無効にできます。

- 汎用USBリダイレクトを使って手動で最適化されたデバイスをリダイレクトするには、[基本設定] ダイアログボックスの [デバイス] タブから [汎用に切り替え] を選択します。
- 汎用USBリダイレクトを使って最適化されたデバイスを自動的にリダイレクトするには、ストレージデバイスの自動リダイレクトを設定し (たとえばAutoRedirectStorage = 1) USBユーザー基本設定を自動接続USBデバイスに設定します。詳しくは、[CTX123015](#)を参照してください。

例：

- 次の例は、ベンダーID (VID) および製品ID (PID) に対する管理者定義のUSBポリシー規則です。  
Allow: VID=046D PID=C626 # Allow Logitech SpaceNavigator 3D Mouse Deny: VID=046D # Deny all Logitech products
- 次の例は、クラス、サブクラス、およびプロトコルに対する管理者定義のUSBポリシー規則です。

Deny: Class=EF SubClass=01 Prot=01 # Deny MS Active Sync devices Allow: Class=EF SubClass=01 # Allow Sync devices Allow: Class=EF # Allow all USB-Miscellaneous devices

USBサポート機能が有効になっており、USB関連のユーザー設定でUSBデバイスに自動接続するように設定されている場合は、USBデバイスが自動的にリダイレクトされます。また、デスクトップアプライアンスモードを使用している場合や、デスクトップでホストされるアプリケーションを実行している場合も、USBデバイスが自動的にリダイレクトされます。ただし、必ずしもすべてのUSBデバイスをリダイレクトするのが適切だとは限りません。詳しくは、[CTX123015](#)を参照してください。

Desktop Viewerのユーザーは、USBデバイスの一覧からデバイスを選択することによって、自動的にリダイレクトされないデバイスを明示的にリダイレクトすることができます。USBデバイスを一覧に表示しないようにしたり、USBデバイスをリダイレクトできないようにしたりするには、ユーザーデバイスおよびVDA上でデバイス規則を定義します。

ReceiverおよびVDAのUSBデバイスリダイレクト規則でリダイレクトするUSBデバイスの種類を指定すると、デフォルトのUSBポリシー規則よりも優先されます。デバイス規則は、ReceiverおよびVDAの両方に適用されます。デバイスが正しくリダイレクトされるように、両方の規則を変更する必要があります。

- ユーザーデバイスのレジストリ（またはReceiver for Linuxの場合はiniファイル）を編集します。インストールメディアに収録されている管理テンプレート（ADMファイル、`\os\lang\Support\Configuration\icaclient_usb.adm`）により、Active Directoryのグループポリシーを使用してユーザーデバイスを変更できます。
- サーバーOSマシン上のVDAレジストリにある、管理者による上書き規則を編集します。インストールメディアに収録されているADMファイル（`\os\lang\Support\Configuration\vda_usb.adm`）により、Active Directoryのグループポリシーを使用してVDAを変更できます。

## 警告

レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があります。Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

製品のデフォルトの規則は、`HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\PortICA\GenericUSB\DeviceRules`に格納されています。このデフォルトの規則は変更しないでください。ただし、以下で説明しているように、製品のデフォルトの規則を参照して管理者による上書き規則を作成できます。管理者による上書き規則は、製品のデフォルトの規則よりも先に評価されます。

管理者による上書き規則は、`HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Citrix\PortICA\GenericUSB\DeviceRules`に格納されています。GPOポリシー規則は、`{Allow|Deny}`の後にスペースで区切った一連の=式の形式で設定します。以下のタグがサポートされます。

| タグ       | 説明                                                                                                                   |
|----------|----------------------------------------------------------------------------------------------------------------------|
| VID      | デバイス記述子のベンダーID                                                                                                       |
| PID      | デバイス記述子の製品ID                                                                                                         |
| REL      | デバイス記述子のリリースID                                                                                                       |
| Class    | デバイス記述子またはインターフェイス記述子のクラス。USB Webサイト ( <a href="http://www.usb.org/">http://www.usb.org/</a> ) にあるUSBクラスコードを参照してください |
| SubClass | デバイス記述子またはインターフェイス記述子のサブクラス                                                                                          |
| Prot     | デバイス記述子またはインターフェイス記述子のプロトコル                                                                                          |

新しいポリシー規則を作成する場合、以下の点に注意してください。

- 大文字と小文字は区別されません。
- 規則の末尾に、#で始まる任意のコメントを追加できます。区切り文字は不要で、コメントは無視されます。
- 空白行およびコメントのみの行は無視されます。
- 区切り文字にはスペースが使用されますが、番号または識別子の間には使用できません。たとえば、`Deny: Class = 08 SubClass=05`は有効ですが、「`Deny: Class=0 Sub Class=05`」は無効です。
- タグには等号(=)を使用する必要があります (MID=1230など)。
- 各規則を1行ずつ記述するか、同一行に記述する場合はセミコロンで区切られたリスト形式である必要があります。

## Important

ADMテンプレートを使用する場合は、規則を単一行に（セミコロン区切りのリストとして）作成する必要があります。

マストレージデバイスなど、最適化されたデバイスを使用する場合は、一般的にポリシーの規則を使用せず、専用のCDMチャネルを使用してデバイスをリダイレクトします。ただし、これらについては次のいずれか方法で無効にできます。

- 汎用USBリダイレクトを使って手動で最適化されたデバイスをリダイレクトするには、[基本設定] ダイアログボックスの [デバイス] タブから [汎用に切り替え] を選択します。
- 汎用USBリダイレクトを使って最適化されたデバイスを自動的にリダイレクトするには、ストレージデバイスの自動リダイレクトを設定し（たとえばAutoRedirectStorage = 1）USBユーザー基本設定を自動接続USBデバイスに設定します。詳しくは、[CTX123015](#)を参照してください。

例：

- 次の例は、ベンダーID (VID) および製品ID (PID) に対する管理者定義のUSBポリシー規則です。

Allow: VID=046D PID=C626 # Allow Logitech SpaceNavigator 3D Mouse Deny: VID=046D # Deny all Logitech products

- 次の例は、クラス、サブクラス、およびプロトコルに対する管理者定義のUSBポリシー規則です。

Deny: Class=EF SubClass=01 Prot=01 # Deny MS Active Sync devices Allow: Class=EF SubClass=01 # Allow Sync devices Allow: Class=EF # Allow all USB-Miscellaneous devices

ユーザーは、仮想セッションの開始前および開始後にUSBデバイスを装着できます。

Receiver for Windowsでは、以下の点について考慮してください。

- セッションを開始した後で装着したデバイスは、Desktop Viewerの [USB] メニューに追加されます。
- USBデバイスが正しくリダイレクトされない場合、仮想セッションが開始されてからデバイスを装着することで問題が解決される場合があります。
- データの損失を避けるため、Windowsで推奨される手順（ [ハードウェアの安全な取り外し] アイコンの使用など）に従ってUSBデバイスを取り外してください。

マストレージデバイスの場合はクライアント側ドライブのマッピング機能が適用され、ユーザーのログオン時にユーザーデバイス上のマストレージデバイスが自動的に仮想デスクトップ上のドライブ文字にマップされます。これらのドライブは、マップされたドライブ文字を持つ共有フォルダーとして表示されます。クライアント側ドライブのマッピングを構成するには、ポリシーの [ICA] > [ファイルリダイレクト]カテゴリの [クライアント側リムーバブルドライブ]設定を使用します。

クライアント側リムーバブルドライブマッピングとUSBサポートの2つの設定項目の主な違いは以下のとおりです。

| 機能                   | クライアント側ドライブのマッピング | 汎用USBリダイレクト                        |
|----------------------|-------------------|------------------------------------|
| デフォルトで有効。            | はい                | いいえ                                |
| 読み取り専用アクセスの構成が可能     | はい                | いいえ                                |
| セッション中にデバイスを安全に取り外せる | いいえ               | はい（ユーザーがオペレーティングシステムで推奨される手順に従う場合） |

汎用USBリダイレクトとクライアント側ドライブのマッピングのポリシーの両方が有効な場合、セッション開始前または後に装着されたマストレージデバイスがクライアント側ドライブのマッピングによりリダイレクトされます。汎用USBリダイレクトとクライアント側ドライブのマッピングのポリシーの両方が有効で、自動リダイレクト (<http://support.citrix.com/article/CTX123015>を参照) が構成されている場合、セッション開始前または後に装着されたマストレージデバイスが汎用USBリダイレクトによりリダイレクトされます。

## 注意

USBリダイレクトは50Kbpsなどの低帯域幅接続でサポートされますが、大きなファイルのコピーはできません。

管理者は、ユーザーが仮想環境のファイルをユーザーデバイス上にコピーすることを許可したり禁止したりできます。デフォルトでは、マップされたクライアント側ドライブ上のフォルダーやファイルに対するセッション内での読み取りや書き込みが許可されます。

マップされたクライアント側ドライブ上のフォルダーやファイルの追加や変更を禁止するには、ポリシーの[クライアント側ドライブへの読み取り専用アクセス]設定を有効にします。この設定項目をポリシーに追加するときは、 [クライアントドライブリダイレクト]設定も追加されており、 [許可] が選択されていることを確認してください。

# 監視

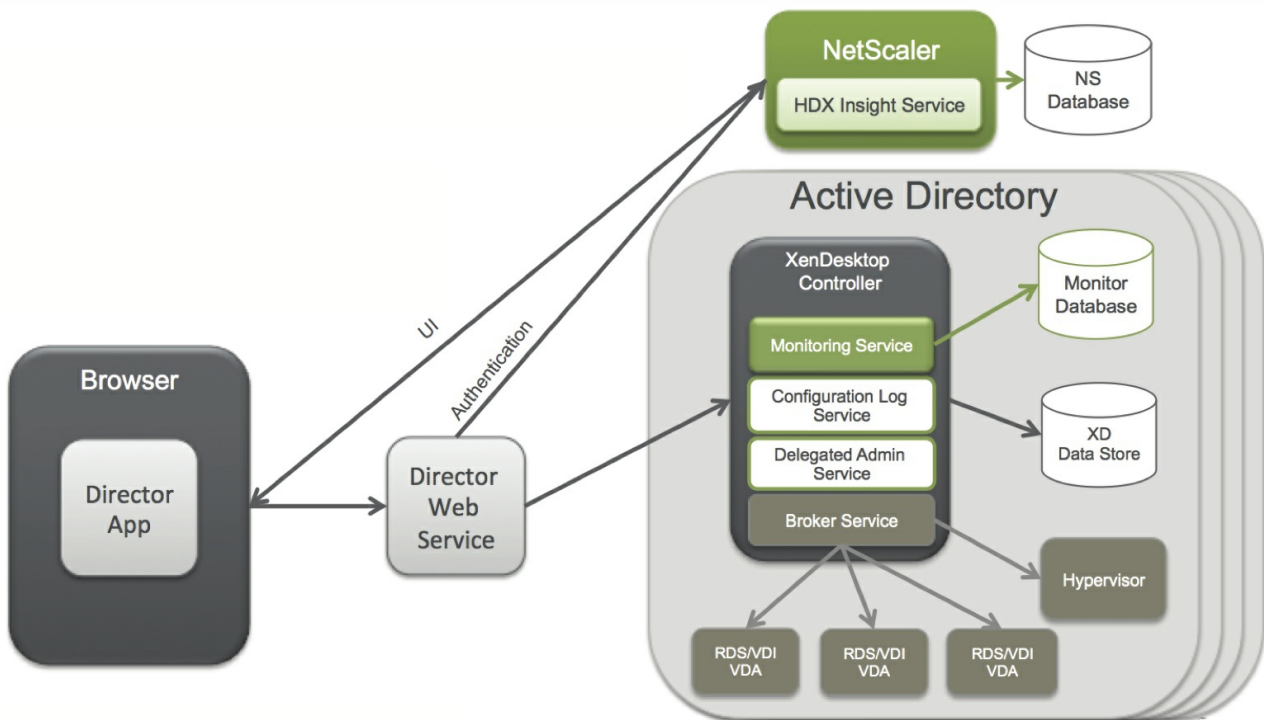
Jan 04, 2017

管理者やヘルプデスク担当者は、Directorから構成ログデータベースにアクセスしたり、Monitor Service APIでODataプロトコルを使用したりしてXenAppおよびXenDesktopのサイトを監視できます。

以下の情報を監視できます。

- セッションの使用
- ログオン処理のパフォーマンス
- 接続およびマシンの障害
- 負荷評価
- 履歴傾向
- インフラストラクチャ
- ユーザーセッション
- マシン

リアルタイムWebツールであるDirectorを使用すると、管理者はエンドユーザーのセッションを監視したり、トラブルシューティングを施したり、そのほかのサポートタスクを実行したりできます。



Directorでは、以下の情報にアクセスできます。

- Broker Agentからのリアルタイムデータ。EdgeSightのPerformance ManagerとNetwork Inspectorの機能が統合されたコンソールを使用します。
- EdgeSightには、ヘルスおよびキャパシティのチェック機能と、NetScaler HDX Insightによる履歴傾向とネットワーク解析機能が含まれており、XenApp/XenDesktop環境のネットワークによるボトルネックを検出できます。
- 監視データベースに格納される履歴データ。構成ログデータベースへのアクセスで使用されます。



- NetScaler GatewayからのICAデータ。HDX Insightが使用されます。
  - XenApp/XenDesktop環境の仮想アプリケーションやデスクトップを使用するエンドユーザーのユーザーエクスペリエンスを視覚化できます。
  - ネットワークデータをアプリケーションデータやリアルタイム測定値に関連付けて効率的にトラブルシューティングを施せます。
  - XenDesktop 7 Directorの監視ツールに統合されています。
- Personal vDiskデータ。ディスク割り当てのランタイム監視とヘルプデスクIT担当者によるPersonal vDiskのリセットが可能になります。
  - コマンドラインツールCtxPvdDiag.exeを使用すると、ユーザーのログ情報が単一ファイルに収集され、それをトラブルシューティングに役立てることができます。

Directorでは、XenApp/XenDesktopサイトのリアルタイムヘルス監視を提供するトラブルシューティングダッシュボードが使用されます。この機能により、管理者がリアルタイムで問題を確認して、エンドユーザーがどのような問題に直面しているかを判断できるようになります。

Session Recordingを使用すると、接続の種類を問わず、企業ポリシーおよび既定のコンプライアンスを遵守するXenAppを実行する任意のサーバー上の任意のユーザーセッションで行われる、画面上のアクティビティを録画することができます。Session Recordingにより録画、カタログ化、およびアーカイブされるセッションを、後で取得して再生することができます。

Session Recordingでは柔軟性の高いポリシーを使用して、アプリケーションセッションの録画を自動的に起動します。IT部門ではSession Recordingを使用して、財務業務や病院での患者情報のシステムなどでエンドユーザーがアプリケーションをどのように使用するかを調査できます。内部統制をサポートすることによって、法規制の順守を徹底し、セキュリティ監視を成功に導きます。同様に、技術サポート部門でこの機能を使用すれば、問題の特定と解決までの時間を短縮することができます。

構成ログ機能では、XenApp/XenDesktopサイトで管理者が行った変更内容が記録されます。構成を変更した後で問題が発生した場合は、構成ログを確認して問題の内容を診断し、トラブルシューティングを施します。また、変更管理、構成の記録、および管理アクティビティのレポート生成が可能です。

構成ログの内容は、Directorの [傾向] ビューで確認できます。このインターフェイスでは、XenDesktop Citrix Studioにアクセスできない管理者に構成変更についての通知を提供できます。

[傾向] ビューでは、特定の期間に行われた構成変更の履歴データを表示できます。これにより、どのような変更がいつ、どのように行われたかを確認して、問題の原因究明に役立てることができます。このビューでは、以下の3つのカテゴリの構成情報が表示されます。

- 接続エラー
- 失敗したデスクトップOSマシン
- 失敗したサーバーOSマシン

管理者は、サイトのMonitor Service APIでODataプロトコルを使用して履歴データを検索できます。これにより、IT担当者に履歴傾向を分析して、プランニングやトラブルシューティングを行えます。また、データを抽出してほかのツールやプロセスで処理することもできます。

Monitor Serviceスキーマでは、以下の種類のデータが提供されます。

- 接続エラーに関するデータ
- エラー状態のマシン
- セッションの使用
- ログオン処理時間
- 負荷分散データ

- [Director](#)
- [Session Recording - XenApp 7.6 FP1およびFP2](#)
- [Personal vDiskの監視](#)
- [構成ログ](#)
- [Monitor Service OData API](#)

# Director

Jan 04, 2017

Directorでは、管理者ごとに異なるインターフェイス（ビュー）が表示されます。Citrix管理者の権限により、表示される内容と実行できるコマンドが異なります。

たとえば、ヘルプデスク管理者にはヘルプデスクタスク用のインターフェイスが表示されます。ヘルプデスク管理者は、問題を報告しているユーザーをDirectorで検索し、アプリケーションやプロセスの状態など、そのユーザーに関するアクティビティを表示できます。ヘルプデスク管理者は応答しないアプリケーションやプロセスを終了したり、ユーザーのマシン上の操作をシャドウしたり、マシンを再起動したり、ユーザープロファイルを再設定したりして問題を解決できます。

これに対して、すべての管理タスクの実行権限を持つ管理者はサイト全体を表示および管理でき、複数のユーザーやマシンに対してコマンドを実行できます。Dashboardには、セッションの状態、ユーザーのログオン、およびサイトインフラストラクチャなど、展開の主要要素に関する概要が表示されます。情報は1分ごとに更新されます。問題が発生すると、発生した問題の数や種類に関する詳細が自動的に表示されます。

Directorは、Delivery Controller上のWebサイトとしてデフォルトでインストールされます。必須要件およびそのほかの詳細については、このリリースのドキュメントの「システム要件」を参照してください。

このリリースのDirectorは、Version 6.5よりも前のXenApp展開やVersion 7よりも前のXenDesktop展開をサポートしません。

Directorで複数のサイトを監視する場合は、Controller、Director、およびそのほかのコアコンポーネントが動作すべてのサーバーのシステムクロックが同期している必要があります。システムクロックが同期していない場合、Directorにサイトの情報が正しく表示されないことがあります。

ヒント：XenApp 6.5ファームと、XenApp 7.5またはXenDesktop 7.xサイトの両方を監視する場合は、XenApp 6.5ファームを監視するためのDirectorコンソールとは別のサーバー上に新しいバージョンのDirectorをインストールすることをお勧めします。

重要：ユーザー名とパスワードがプレーンテキストで送信されないように、Director接続ではHTTPではなくHTTPSでの接続のみを許可することを強く推奨します。特定のツールを使用すると、HTTP（非暗号化）ネットワークパケット内のプレーンテキストのユーザー名やパスワードを読み取ることができるため、ユーザーにとってセキュリティ上のリスクとなる場合があります。

## 権限を構成するには

Directorにログオンする管理者は、Active Directoryドメインユーザーで、以下の権限を持っている必要があります。

- 検索するすべてのActive Directoryフォレストを読み取る権限（「[詳細構成](#)」を参照）
- 構成済みの委任管理者の役割（「[管理権限の委任とDirector](#)」を参照）。
- ユーザーをシャドウするには、WindowsリモートアシスタンスのMicrosoftグループポリシーを使って管理者を構成する必要があります。また、次の点に注意してください。
  - VDAをインストールするすべてのユーザーデバイス上で、Windowsリモートアシスタンス機能が有効である必要があります。この機能は、デフォルトで有効になっています。
  - Directorをインストールするサーバーに、Windowsリモートアシスタンス機能がインストールされている必要があります。この機能は、デフォルトでインストールされています。ただし、デフォルトでは無効になっています。Directorを使ってエンドユーザーを支援する場合、この機能を有効にする必要はありません。セキュリティ上の理由から、この機能を無効にしておくことをお勧めします。
  - 管理者がWindowsリモートアシスタンスを開始できるようにするには、適切なMicrosoftグループポリシー設定を使用し

て管理者に必要な権限を付与します。詳しくは、「[CTX127389 : Desktop Directorに対してリモートアシスタンスを有効にする方法](#)」を参照してください。

- XenDesktop 7よりも前のバージョンのVDAがインストールされたユーザーデバイスについては、追加の構成が必要です。「[XenDesktop 7よりも前のVDAに対する権限の構成](#)」を参照してください。

## Directorをインストールするには

注：Directorがファーム内のすべてのXenAppワーカーを見つけることができるようにするには、ファームによって使用されるDNSサーバー上にXenAppサーバーがあるサブネットにリバースDNSゾーンを追加する必要があります。

Directorはインストーラーを使ってインストールします。このインストーラーは、必要なサードパーティ製ソフトウェアやコンポーネントをインストールして、DirectorのWebサイトをセットアップし、基本的な構成を行います。インストーラーによるデフォルトの構成のままで、一般的な展開を管理できます。インストール時にDirectorを含めなかった場合は、インストーラーを再度実行してDirectorをインストールします。追加のコンポーネントをインストールするには、インストーラーを再度実行して必要なコンポーネントを選択します。インストーラーの使用については、インストールに関するドキュメントを参照してください。個々のMSIファイルではなく、製品のメインインストーラーを実行して各コンポーネントをインストールすることをお勧めします。

Controller上にインストールされたDirectorでは、サーバーアドレスとしてlocalhostが自動的に構成され、DirectorはデフォルトでローカルのControllerと通信します。

Controllerとは別の専用サーバー上にDirectorをインストールする場合は、Controllerの完全修飾ドメイン名 (FQDN) またはIPアドレスを指定する必要があります。Directorは、ここで指定したアドレスのControllerと通信します。監視する各サイトについてControllerのアドレスを1つずつ指定します。各サイトにあるほかのすべてのControllerは自動的に検出され、指定したControllerにエラーが発生した場合はほかのControllerにフォールバックされます。

注：Directorは、Controller間で負荷分散を行いません。

WebブラウザとWebサーバー間の通信を保護するため、DirectorをホストするIIS WebサイトでSSLを実装することをお勧めします。この手順については、Microsoft IISのドキュメントを参照してください。Director側では、SSLを有効にするために何らかの構成を行う必要はありません。

## Director 7.6.300をインストールするには

Director 7.6.300はFramehawk仮想チャネルをサポートし、最新の製品修正プログラムを配信します。これらは、XenAppおよびXenDesktop 7.6 FP2、XenApp 6.5 FP3で入手できます。

注：IISに必要なすべての機能が選択されているかチェックしてください。完全な一覧については、[CTX142260](#)を参照してください。

1. Directorをダウンロードし、MSIファイルであるDesktopDirector.MSIまたはDesktopDirector\_x64.MSIを実行します。
2. XenAppおよびXenDesktopインストールメディアのCitrix PolicyフォルダーにあるCitrixGroupPolicyManagement.MSIをインストールします。
3. Delivery ControllerでDirectorを構成します。これにはC:\inetpub\wwwroot\Director\toolsにあるDirectorConfig.exeツールを使用します。詳しくは、[CTX137990](#)を参照してください。
4. ASP.netをIISに登録します。これを実行するには、コマンドC:\inetpub\wwwroot\Director\tools>DirectorConfig.exe /registerdotnet実行します。
5. VDA上のWMIProxyをインストールまたはアップグレードし、マシンを再起動します。WMIProxy\_x64.MSIおよびWMIProxy\_x86.MSIはDirector 7.6.300のダウンロードファイルに含まれています。注：WMIProxyがアップグレードされていない場合、Framehawk仮想チャネルはFramehawkが有効になったVDAで「非互換」として表示されます。
6. XDPoshSnapin\_Hot fixをDelivery Controllerでアップグレードし、次にStudioを再起動します。XDPoshSnapin MSIは手順1でダウンロードしたDirectorファイルから実行できます。これは、委任管理者またはカスタム管理者がFramehawk仮想

チャンネル情報を見るのに必要です。

## Directorにログオンするには

DirectorのWebサイトは、https または「http://Directorにあります。

複数サイト環境でいずれかのサイトがダウンしている場合、Directorへのログオンに時間がかかる場合があります。これは、ダウンしているサイトへの接続が試行されるためです。

## XenApp 6.5用のDirectorをインストールするには

XenDesktop用のDirectorが既にインストールされている場合は、以下のXenApp用の構成を行います。

- 「[詳細構成](#)」の「サイトをDirectorに追加するには」の説明に従って、各Directorサーバー上のIIS管理コンソールで [アプリケーションの設定] のXenAppサーバーアドレスの一覧を更新します。各XenAppファームについて1つのコントローラーのサーバーアドレスを指定します。XenAppファームのほかのコントローラーは自動的にフェールオーバー用に使用されません。Directorは、コントローラー間で負荷分散を行いません。
- 「[権限の構成](#)」で説明されているように、各XenAppワーカーサーバーがWinRMクエリを受け入れるように構成します。
- DirectorとXenApp間の通信で使用されるポート2513に対するファイアウォール例外を構成します。

## XenApp 6.5用のDirectorを新規インストールするには

XenApp 6.5用のDirectorを新規にインストールするには、次の手順に従います。通常、DirectorはXenAppコントローラーとは別のコンピューター上にインストールします。

1. XenApp 7.6のインストールメディアから、Directorをインストールします。
2. 「[詳細構成](#)」の「サイトをDirectorに追加するには」の説明に従って、各Directorサーバー上のIIS管理コンソールで [アプリケーションの設定] のXenAppサーバーアドレスの一覧を更新します。  
各XenAppサイトについて1つのコントローラーのサーバーアドレスを指定します。XenAppサイトのほかのコントローラーは自動的にフェールオーバー用に使用されます。Directorは、コントローラー間で負荷分散を行いません。

**重要：** XenAppアドレスはService.AutoDiscoveryAddressesXAに指定し、デフォルト設定のService.AutoDiscoveryAddressesに指定しないでください。

3. Director WMIプロバイダーのインストーラーがDVDのSupport\DirectorWMIProviderフォルダーに収録されています。適切なすべてのXenAppサーバー（コントローラーおよびセッションを実行するワーカー）上にこれをインストールします。Ifwinrmが構成されていない場合は、winrm qcコマンドを実行します。
4. 「[権限の構成](#)」で説明されているように、各XenAppワーカーサーバーがWinRMクエリを受け入れるように構成します。
5. DirectorとXenApp間の通信で使用されるポート2513に対するファイアウォール例外を構成します。
6. WebブラウザとWebサーバー間の通信を保護するため、DirectorをホストするIIS WebサイトでSSLを実装することをお勧めします。  
この手順については、Microsoft IISのドキュメントを参照してください。SSLを有効にするためにDirectorを構成する必要はありません。

# 管理権限の委任と Director

Jan 04, 2017

管理権限の委任機能では、管理者、役割、およびスコープという3つの概念が使用されます。管理者の権限は、その管理者の役割とそのスコープに基づいて定義されます。たとえば、管理者にヘルプデスク管理者の役割を割り当てて、その役割のスコープとして特定のサイトのエンドユーザーを指定できます。

委任管理者の作成について詳しくは、「[管理権限の委任](#)」を参照してください。

付与されている管理権限により、その管理者に表示される Director のインターフェイスと実行可能なタスクが決定されます。権限により、次の内容が決定されます。

- その管理者がアクセスできる Director の表示内容。これを「ビュー」と呼びます。
- その管理者が表示したり操作したりできるデスクトップ、マシン、およびセッション。
- ユーザーセッションのシャドウやメンテナンスモードの有効化など、その管理者が実行できるコマンド。

組み込みの役割および権限によっても、管理者が Director で実行できるタスクが決定されます。

| 管理者の役割            | Directorでの権限                                                                                                                                                                                                                                  |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| すべての管理権限を実行できる管理者 | すべてのビューに制限なくアクセスして、ユーザーセッションのシャドウ、メンテナンスモードの有効化、傾向データのエクスポートなどすべてのコマンドを実行できます。                                                                                                                                                                |
| デリバリーグループ管理者      | すべてのビューに制限なくアクセスして、ユーザーセッションのシャドウ、メンテナンスモードの有効化、傾向データのエクスポートなどすべてのコマンドを実行できます。                                                                                                                                                                |
| 読み取り専用管理者         | すべてのビューに制限なくアクセスして、一般的な情報と、指定されているスコープのすべてのオブジェクトを表示できます。HDXチャネルからレポートをダウンロードして、[傾向] ビューのエクスポートオプションを使って傾向データをエクスポートできます。そのほかのコマンドは実行できず、ビューで設定を変更することはできません。                                                                                 |
| ヘルプデスク管理者         | [ヘルプデスク] および [ユーザーの詳細] ビューにのみアクセスでき、委任されたオブジェクトのみを表示できます。ユーザーセッションをシャドウしたり、そのユーザーに対してコマンドを実行したりできます。メンテナンスモードを有効にしたり解除したりできます。デスクトップOSマシンの電源制御オプションを使用できます。<br>[ダッシュボード] ビュー、[傾向] ビュー、および [フィルター] ビューにはアクセスできません。サーバーOSマシンの電源制御オプションは使用できません。 |
| マシンカタログ管理者        | アクセスできません。この管理者は、Directorを使用したりデータを表示したりできません。マシン詳細ページ (マシンベースの検索) にはアクセスできます。                                                                                                                                                                |
|                   |                                                                                                                                                                                                                                               |

ホスト管理  
管理者の  
役割

アクセスできません。この管理者は、Directorを使用したりデータを表示したりできません。

## Director管理者のカスタム役割を構成するには

Studioでは、組織の要件に応じてDirector用のカスタムの役割を構成して、管理権限を柔軟に委任できます。たとえば、組み込みのヘルプデスク管理者の役割を制限して、この管理者がユーザーのセッションをログオフすることを禁止できます。

Director用のカスタムの役割を作成する場合は、その役割に以下の一般的な権限も付与する必要があります。

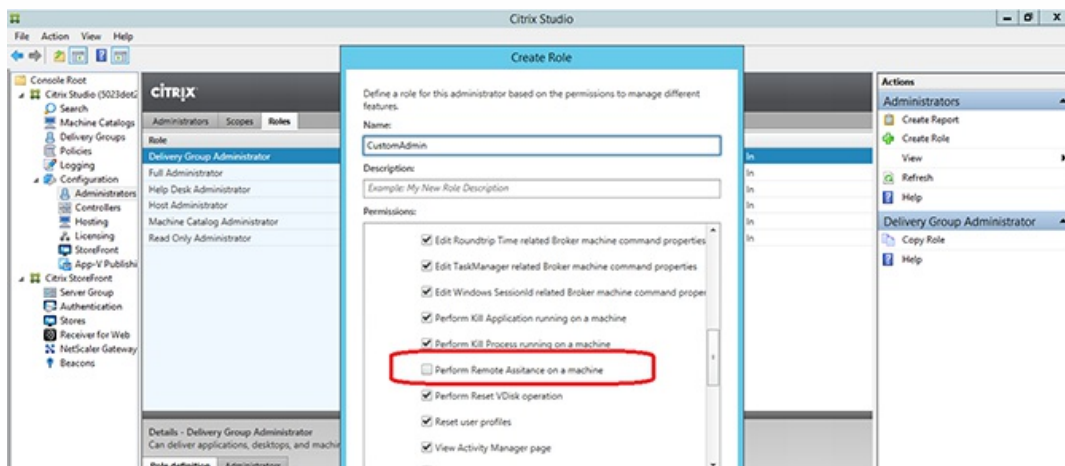
- DirectorにログオンするためのDelivery Controller権限。
- デリバリーグループのデータをDirectorで閲覧するための権限。

または、既存の役割をコピーしてカスタムの役割を作成し、異なるビューのための権限を追加することができます。たとえば、ヘルプデスクの役割をコピーして、[ダッシュボード] ページや [フィルター] ページを表示するための権限を追加できます。

以下のDirector用の権限を追加します。

- マシンで実行中のアプリケーションの強制終了
- マシンで実行中のプロセスの強制終了
- マシン上でのリモートアシスタンス
- vDiskのリセット操作
- ユーザープロファイルのリセット
- クライアント詳細ページの表示
- ダッシュボードページの表示
- フィルターページの表示
- マシン詳細ページの表示
- 傾向ページの表示
- ユーザー詳細ページの表示

この例では、シャドウ機能（マシン上でのリモートアシスタンス）が無効になっています。



さらに、ほかのコンポーネントの権限の一覧から、次の権限の追加を検討します。

- デリバリーグループの権限：
  - デリバリーグループメンバーシップによるマシンのメンテナンスモードの有効/無効
  - デリバリーグループメンバーシップによるWindowsデスクトップマシンの電源操作

- デリバリーグループメンバーシップによるマシンのセッション管理



# XenDesktop 7よりも前のVDAに対する権限の構成

Jan 04, 2017

ユーザーデバイス上にXenDesktop 7よりも前のバージョンのVDAがインストールされている場合、Directorには展開からの情報に加えて、Windowsリモート管理 (WinRM) によるリアルタイム状態と測定値が提供されます。

さらに、この手順を使用してXenDesktop 5.6 Feature Pack1のリモートPC用にWinRMを構成します。

デフォルトでは、デスクトップマシンのローカル管理者 (一般的にはドメイン管理者および特権のあるそのほかのユーザー) だけが、リアルタイムデータを表示するための権限を持っています。

WinRMのインストールと構成については、[CTX127390](#)を参照してください。

ほかのユーザーがリアルタイムデータを確認できるようにするには、そのユーザーに権限を付与する必要があります。たとえば、HelpDeskUsersというActive Directoryセキュリティグループのメンバーである複数のDirectorユーザー (HelpDeskUserA、HelpDeskUserBなど) がいるとします。グループにはStudioでヘルプデスク管理者の役割が割り当てられており、必要なDelivery Controller権限が付与されています。ただし、このグループはデスクトップマシンからの情報にもアクセスする必要があります。

この場合、次のいずれかの方法で必要な権限を構成できます。

- Directorユーザーに権限を付与する (偽装モデル)
- Directorサービスに権限を付与する (信頼されたサブシステムモデル)

## Directorユーザーに権限を付与するには (偽装モデル)

Directorでは、デフォルトで偽装モデルが使用されます。つまり、デスクトップマシンへのWinRM接続は、DirectorユーザーのIDを使って実行されます。つまり、このユーザーにはデスクトップに対する適切な権限が必要です。

必要な権限は、次のいずれかの方法で構成できます (これらの方法について詳しくは後述します)。

1. ユーザーをデスクトップマシン上のローカル管理者グループに追加する。
2. Directorで必要な特定の権限をユーザーに付与する。この方法では、デスクトップマシンのすべての管理権限をDirectorユーザー (HelpDeskUsersグループなど) に付与しなくても、必要な権限だけを付与できます。

## Directorサービスに権限を付与するには (信頼されたサブシステムモデル)

Directorユーザーにデスクトップマシン上の権限を付与する代わりに、WinRM接続にサービスIDが使用されるようにして、そのサービスIDに適切な権限のみを付与できます。

このモデルでは、DirectorのユーザーにはWinRMを呼び出す権限を付与しません。ユーザーはDirectorを使ってデータにアクセスするだけです。

IISのDirectorアプリケーションプールは、サービスIDとして実行するように構成されています。デフォルトでは、APPPOOL\Director仮想アカウントが使用されます。リモート接続を実行すると、このアカウントがサーバーのActive Directoryコンピューターアカウント (MyDomain\DirectorServer\$など) として表示されます。このアカウントに適切な権限を付与する必要があります。

複数のDirector Webサイトが展開されている環境では、各Webサーバーのコンピューターアカウントを、適切な権限で構成されているActive Directoryセキュリティグループに追加する必要があります。

WinRMに対してユーザーのIDではなくサービスIDを使用するようにDirectorを設定するには、[詳細構成](#)の説明に従って

次の設定を構成します。

Service.Connector.WinRM.Identity = Service

必要な権限は、次のいずれかの方法で構成できます。

1. サービスアカウントをデスクトップマシン上のローカル管理者グループに追加する。
2. Directorで必要な特定の権限をサービスアカウントに付与する（後述）。この方法では、デスクトップマシンのすべての管理権限をサービスアカウントに付与しなくても、必要な権限だけを付与できます。

特定のユーザーまたはグループに権限を割り当てるには

DirectorがWinRMでデスクトップマシンからの情報にアクセスできるようにするには、以下の権限が必要です。

- WinRM RootSDDLの読み取りおよび実行権限
- 以下のWMI名前空間権限：
  - root/cimv2 — リモートアクセス
  - root/citrix — リモートアクセス
  - root/RSOP — リモートアクセスおよび実行
- 以下のローカルグループのメンバーシップ：
  - Performance Monitor Users
  - Event Log Readers

これらの権限を自動的に付与するには、インストールメディアのx86\Virtual Desktop Agentおよびx64\Virtual Desktop Agentフォルダーおよびtoolsフォルダーに収録されているConfigRemoteMgmt.exeツールを使用します。すべてのDirectorユーザーに上記の権限を付与する必要があります。

Active Directoryセキュリティグループ、ユーザー、またはコンピューターアカウントに上記の権限を付与したり、アプリケーションやプロセスを終了するための権限を付与したりするには、管理特権でコマンドプロンプトを開き、次の引数を指定してこのツールを実行します。

```
ConfigRemoteMgmt.exe /configwinrmuser <domain\name>
```

ここで、<name>はセキュリティグループ、ユーザー、またはコンピューターアカウントです。

ユーザーのセキュリティグループに必要な権限を付与するには次のコマンドを実行します。

```
ConfigRemoteMgmt.exe /configwinrmuser <domain>\HelpDeskUsers
```

特定のコンピューターアカウントに権限を付与するには次のコマンドを実行します。

```
ConfigRemoteMgmt.exe /configwinrmuser <domain>\DirectorServer$
```

アプリケーションやプロセスを終了したり、シャドウ機能を使用したりする権限を付与するには次のコマンドを実行します。

```
ConfigRemoteMgmt.exe /configwinrmuser <domain\name> /all
```

特定のユーザーグループに権限を付与するには次のコマンドを実行します。

```
ConfigRemoteMgmt.exe /configwinrmuser <domain>\HelpDeskUsers /all
```

このツールのオンラインヘルプを表示するには次のコマンドを実行します。

```
ConfigRemoteMgmt.exe
```

# HDX Insightの構成

Jan 04, 2017

注：この機能は、組織のライセンスおよび管理者権限によっては使用できない場合があります。

HDX Insightにより、EdgeSightのネットワーク分析およびパフォーマンス管理機能がDirectorに統合されます。

- EdgeSightネットワーク分析機能では、HDX Insightを使用してネットワークのアプリケーションおよびデスクトップのコンテキストビューを提供します。これにより、環境内のICAトラフィックをDirectorで詳細に分析できます。
- EdgeSightのパフォーマンス管理機能により、履歴保持および傾向に関するレポートを生成できます。データの履歴保持とリアルタイム評価により、管理者はサーバーのキャパシティとヘルスに関する傾向レポートを作成できます。

Directorでこの機能を有効にすると、HDX Insightにより以下の追加情報がDirectorに提供されます。

- [傾向] ページには、展開環境全体におけるアプリケーション、デスクトップ、ユーザーに対する遅延と帯域幅の影響が表示されます。
- [ユーザーの詳細] ページには、特定のユーザーセッションに特化した遅延と帯域幅情報が表示されます。

## 制限事項

- ICAセッション往復時間 (RTT) には、Receiver for Windows 3.4以降およびReceiver for Mac 11.8以降のデータが正確に表示されます。これらのバージョンよりも前のバージョンのReceiverについては、正確なデータが表示されません。
- [傾向] ビューでは、XenDesktop 7よりも前のバージョンのVDAについてはHDX接続のログオンデータが収集されません。以前のバージョンのVDAについては、チャートデータが0として表示されます。

## Director上でEdgeSight ネットワーク分析機能を構成するには

EdgeSightでは、NetScaler HDX Insightによるネットワーク分析が実行されます。これにより、Citrixアプリケーションおよびデスクトップの管理者は、ネットワークパフォーマンスを低下させる問題のトラブルシューティングおよび関連付けを行います。

NetScaler Insight CenterをインストールしてDirectorで構成し、EdgeSightネットワーク分析を有効にする必要があります。NetScaler Insight Centerは、Citrix.comからダウンロード可能な仮想マシン (アプライアンス) です。Directorでは、EdgeSightネットワーク分析により、環境のトラフィック情報を収集します。この情報に基づいて、HDX InsightはクライアントとバックエンドのCitrixインフラストラクチャ間のICAプロトコルの高精度な分析機能を提供します。

1. Directorがインストールされているサーバー上のコマンドラインプロンプトで、C:\inetpub\wwwroot\Director\toolsにあるDirectorConfigコマンドに/confignetscalerパラメーターを指定して実行します。
2. 画面上の指示に従って、NetScaler Insight Centerマシン名 (完全修飾ドメイン名またはIPアドレス)、ユーザー名、パスワード、および接続の種類 (HTTPまたはHTTPS) を構成します。
3. 変更を確認するには、いったんログオフして再ログオンします。

# 詳細構成:

Jan 04, 2017

ここでは次のことについて説明します。

[複数のActive Directoryドメインとフォレスト間でユーザーをサポートするには](#)

[サイトをDirectorに追加するには](#)

[アクティビティマネージャーで実行中のアプリケーションを非表示にするには](#)

複数のサイトまたは複数のActive Directoryフォレストのサポートなど、一部の詳細なDirector構成は、インターネットインフォメーションサービス (IIS) マネージャーの設定を介して制御します。

重要：IISの設定を変更すると、Directorサービスが自動的に再起動してユーザーをログオフします。

IISを使って詳細設定を構成するには、次の手順に従います。

1. インターネットインフォメーションサービス (IIS) マネージャーコンソールを開きます。
2. [Default Web Site] ノードを開き、[Director] Webサイトを選択します。
3. [アプリケーションの設定] をダブルクリックします。
4. 編集する設定をダブルクリックします。

Platinumライセンスはデフォルトで90日間データを保持します。構成について詳しくは、[データの粒度と保持](#)を参照してください。

DirectorはActive Directoryを使ってユーザーを検索し、ユーザーおよびマシンの追加情報を照会します。Directorのデフォルトでは、以下のドメインまたはフォレストが検索されます。

- 管理者のアカウント属しているドメインやフォレスト。
- DirectorのWebサーバーが属しているドメインやフォレスト (管理者が属しているものと異なる場合)。

Directorでは、Active Directoryグローバルカタログによるフォレストレベルでの検索が試行されます。管理者にフォレストレベルで検索する権限がない場合、ドメインのみが検索されます。

ほかのActive Directoryドメインまたはフォレストからのデータを検索または照会するには、対象のドメインまたはフォレストを明示的に設定する必要があります。次の設定を構成します。

```
Connector.ActiveDirectory.Domains = (user),(server)
```

値属性userおよびserverは、それぞれDirectorユーザー (つまり管理者) のドメインおよびDirectorサーバーのドメインを表しています。

ほかのドメインまたはフォレストからのデータを検索するには、次のようにドメイン名をリストに追加します。

```
Connector.ActiveDirectory.Domains = (user),(server),,
```

リストに追加した各ドメインについて、Directorによりフォレストレベルの検索が試行されます。管理者にフォレストレベルで検索する権限がない場合、ドメインのみが検索されます。

Directorがインストール済みの場合は、複数のサイトを監視できるように構成できます。これを行うには、各Directorサーバー上でIIS管理コンソールを使って [アプリケーションの設定] のサーバーアドレスの一覧を更新します。

各サイトのControllerのアドレスを次の設定に追加します。

`Service.AutoDiscoveryAddresses = SiteAController,SiteBController`

ここでSiteAControllerおよびSiteBControllerは、2つの異なるサイトのDelivery Controllerのアドレスです。

XenApp 6.5の場合は、各XenAppファームのコントローラーのアドレスを次の設定に追加します。

`Service.AutoDiscoveryAddressesXA = FarmAController,FarmBController`

ここでFarmAControllerおよびFarmBControllerは、2つの異なるファームのXenAppコントローラーのアドレスです。

次の方法でXenApp 6.5ファームのコントローラーを追加することもできます。

`DirectorConfig.exe /xenapp FarmControllerName`

Directorのアクティビティマネージャーのデフォルトでは、そのユーザーのセッションで実行されているすべてのアプリケーションが一覧表示されます。この情報を表示するには、Directorのアクティビティマネージャー機能へのアクセス権限が必要です。この権限を持つ管理者の役割は、すべての管理権限を実行できる管理者、デリバリーグループ管理者、およびヘルプデスク管理者です。

ユーザーのプライバシーと使用しているアプリケーションを保護するため、[アプリケーション] タブでアプリケーションの一覧を非表示にできます。

注意：レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。また、レジストリファイルのバックアップを作成してから、レジストリを編集してください。

1. VDAで、レジストリキー `HKEY_LOCAL_MACHINE\Software\Citrix\Director\TaskManagerDataDisplayed` を変更します。デフォルトでは1に設定されています。値を0に変更すると、アクティビティマネージャーに情報が表示されなくなります。
2. Directorがインストールされたサーバー上で、実行中のアプリケーションの表示を制御する設定を変更します。デフォルトの値はtrueで、これにより [アプリケーション] タブに実行中のアプリケーションの一覧が表示されます。値をfalseに変更すると、アプリケーションの一覧が表示されなくなります。このオプションは、VDAではなくDirectorのアクティビティマネージャーにのみ適用されます。

次の設定の値を変更します。

`UI.TaskManager.EnableApplications = false`

重要：実行中のアプリケーションの表示を無効にするには、これらの両方の値を変更してください。

# 展開環境の監視

Jan 19, 2017

すべての管理権限を実行できる管理者としてDirector起動すると、サイトのヘルス状態や使用状況を監視するための[ダッシュボード]が開きます。

直近の60分間にエラーが発生していない場合、各パネルは閉じています。エラーが発生している場合はそのエラーを示すパネルが自動的に開きます。

注：組織のライセンスおよび管理者権限によって、表示されるオプションや機能は異なります。

| パネル                             | 説明                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ユーザー接続エラー                       | 過去60分間の接続エラーが表示されます。エラー総数の横にあるカテゴリをクリックして、各種のエラーの測定値を確認します。隣接する表には、発生したエラー数がデリバリーグループごとに表示されます。                                                                                                                                                                                                                                                                                 |
| 失敗したデスクトップOSマシンまたは失敗したサーバーOSマシン | 過去60分間の総エラー数がデリバリーグループごとに表示されます。エラーの種類として、起動の失敗、起動時のスタック、および未登録があります。サーバーOSマシンの場合は、最大負荷に達しているマシンも含まれます。                                                                                                                                                                                                                                                                         |
| ライセンスの状態                        | <ul style="list-style-type: none"><li>ライセンスサーバーアラートには、ライセンスサーバーから送信されたアラートメッセージとそのアラートを解決するための操作が表示されます。</li><li>Delivery Controllerアラートには、Delivery Controllerから送信されたライセンス状態の詳細が表示されます。</li></ul> <p>アラートのしきい値は、Studioで設定できます。</p> <p>ライセンスサーバーのバージョンが11.12.1よりも前の場合や、Delivery ControllerがXenApp 7.6またはXenDesktop 7.6よりも古い場合、ライセンスサーバーアラートとDelivery Controllerアラートは表示されません。</p> |
| 接続セッション                         | すべてのデリバリーグループでの過去60分間の接続セッションが表示されます。                                                                                                                                                                                                                                                                                                                                           |
| 平均ログオン処理時間                      | 過去60分間のログオン処理に関するデータが表示されます。左側にある大きなサイズの数値は、全体的な平均ログオン処理時間を示します。<br>この平均には、XenDesktop 7.0より前のバージョンのVDAへのログオンデータは含まれません。                                                                                                                                                                                                                                                         |
| インフラストラクチャ                      | サイトのホスト、Controller、およびインフラストラクチャのヘルス状態が表示されます。パフォーマンスに関するアラートを確認できます。<br>ホストの場合、接続状態、CPU、メモリ、帯域幅（ネットワーク使用）、およびストレージ（ディスク使用）のヘルス状態がXenServerまたはVMwareからの情報を使って監視されます。<br><br>たとえば、XenCenterでは、サーバーまたは仮想サーバーのCPU、ネットワークI/O、またはディスクI/Oの使用量が特定のしきい値を超えた場合にパフォーマンスアラートが発せられるように構成できます。アラートの送信間隔はデフォルトで60分ですが、必要に応じて変更できます。詳しくは、                                                      |

|     |                                                    |
|-----|----------------------------------------------------|
| パネル | XenServerのドキュメントでパフォーマンスアラートの設定について参照してください。<br>説明 |
|-----|----------------------------------------------------|

注：ホスト上でサポートされていない種類の測定値のアイコンは表示されません。たとえば、System Center Virtual Machine Manager (SCVMM) ホスト、AWS、およびCloudStackを使用する環境では、ヘルス情報が表示されません。以下に示すオプションを使って問題のトラブルシューティングを行います。

- ユーザーマシン電源の制御
- マシンへの接続の無効化

セッションが切断状態になると、セッションおよびアプリケーションは終了しませんが、サーバーとユーザーデバイス間の通信が切断されます。

| 操作                               | 説明                                                                                                                                                                                                  |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ユーザーが接続しているマシンまたはセッションを表示する      | 【アクティビティマネージャー】および【ユーザーの詳細】ビューで、ユーザーが接続しているマシンまたはセッションと、そのユーザーがアクセスしているすべてマシンおよびセッションの一覧を表示します。セッションの一覧にアクセスするには、そのユーザーのビューのタイトルバーにあるセッション切り替え用のアイコンをクリックします。「 <a href="#">セッションの復元</a> 」を参照してください。 |
| すべてのデリバリーグループで接続されたセッションの総数を表示する | 【ダッシュボード】の【接続セッション】ペインには、すべてのデリバリーグループでの過去60分間の接続セッション数が表示されます。その合計数をクリックすると、【フィルター】ビューが開きます。ここでは、デリバリーグループごとのセッションデータや、すべてのデリバリーグループでの特定期間での使用量を視覚的に確認できます。                                        |
| 長期間のデータを表示する                     | 【傾向】ビューの【セッション】タブでは、長期間（つまり直近の60分よりも前）の接続セッションおよび切断セッションに関するデータを詳細に確認できます。この情報を表示するには、【履歴傾向の表示】をクリックします。                                                                                            |

注：Virtual Delivery Agent 7より前のバージョンのVDAを実行する場合、セッションに関する一部の情報がDirectorに表示されません。この場合、【ユーザーの詳細】ビューと【アクティビティマネージャー】パネルに、情報を使用できないことを示すメッセージが表示されます。

【ダッシュボード】で数値をクリックしたり【フィルター】メニューから事前定義のフィルターを選択すると、【フィルター】ビューが開きます。ここには、選択したマシンまたはエラーの種類に関するデータが表示されます。

事前定義のフィルターはそのままでは編集できませんが、それをカスタムフィルターとして保存してから編集することができます。さらに、すべてのデリバリーグループでのマシン、接続、およびセッションのカスタムフィルタービューを作成できます。

1. 以下のビューを選択します。

- マシン — 【デスクトップOSマシン】タブまたは【サーバーOSマシン】タブを選択します。これらのタブには構成されたマシンの数が表示されます。また、【サーバーOSマシン】タブには負荷評価基準インデックスが表示され、その測定値上にマウスポインターを置くと各パフォーマンスカウンターの測定値やセッション数がツールチップとして表示されます。
- セッション — セッション数が表示されます。【マシン】ビューでセッション数を表示することもできます。

- 接続 — 直近の60分、24時間、または7日間の接続が表示されます。
2. [フィルター基準] で特定の条件を定義します。
  3. 必要に応じて、各ビューで追加のタブを使用してフィルターを実行します。
  4. 必要に応じて追加の列を選択して、より詳細な情報を表示します。
  5. フィルターに名前を付けて保存します。  
後でフィルターを開くには、[フィルター] メニューでエラーの種類（マシン、セッション、または接続）を選択し、保存済みのフィルターを選択します。
  6. [マシン] ビューまたは [接続] ビューでは、一覧でマシンを選択して電源制御操作を実行できます。マシンおよび接続エラーに関する失敗の理由および推奨される操作については、[Citrix Director 7.6 Failure Reasons Troubleshooting Guide](#)を参照してください。
  7. [セッション] ビューでは、セッション制御を実行したりメッセージを送信したりできます。

以下に示すオプションを使って問題のトラブルシューティングを行います。

- ユーザーマシン電源の制御
- マシンへの接続の無効化

[傾向] ビューでは、各サイトのセッション、接続エラー、マシン障害、ログオンパフォーマンス、および負荷評価についての履歴傾向情報が表示されます。これらの情報を表示するには、[ダッシュボード] または [フィルター] ビューで [傾向] をクリックします。

ズームインドリルダウン機能により、（グラフ内のデータポイントをクリックして）ある期間について着目し、その傾向に関連する詳細情報を表示させて、傾向チャートを参照できます。これにより、表示中の傾向により誰が、または何が影響を受けているかについてより詳細に把握できます。

各グラフのデフォルトの表示範囲を変更するには、[期間] フィルターを変更して適用します。

| 操作            | 説明                                                                                             |
|---------------|------------------------------------------------------------------------------------------------|
| グラフデータのエクスポート | エクスポートするデータを含むタブを選択します。[エクスポート] をクリックして、エクスポート形式（PDF、Excel、またはCSV）を選択します。                      |
| セッションの傾向の表示   | [セッション] タブから、同時接続セッション数に関するより詳細な情報を表示するデリバリーグループと期間を選択します。                                     |
| 接続エラーの傾向の表示   | [接続エラー] タブで、接続エラー情報を表示するマシンの種類、エラーの種類、デリバリーグループ、および期間を選択します。                                   |
| マシン障害の傾向の表示   | [失敗したデスクトップOSマシン] タブまたは [失敗したサーバーOSマシン] タブで、障害情報を表示するエラーの種類、デリバリーグループ、および期間を選択します。             |
| ログオンパフォーマンス   | [ログオンパフォーマンス] タブで、ログオンパフォーマンス情報（ログオン処理時間およびログオン数）を表示するデリバリーグループ、および期間を選択します。このビューには、仲介処理時間や仮想マ |



|                                |                                                                                                                                                                                                                                                                    |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| スの傾向の表示                        | シンの起動時間などのログオンフェーズにおける平均時間も表示されます。<br>このデータはユーザーのログオンに関するものであり、切断セッションへの再接続は含まれません。                                                                                                                                                                                |
| 負荷評価の傾向の表示                     | [負荷評価基準インデックス] タブで、サーバーOSマシン間で分散された負荷に関する情報を表示します。このグラフでは、対象のデリバリーグループ、サーバーOSマシン、および期間を指定できます。                                                                                                                                                                     |
| ホストされたアプリケーションの使用量の表示          | この機能は、組織のライセンスによっては使用できない場合があります。<br>[ホストされたアプリケーションの使用量] タブでデリバリーグループと期間を選択すると、最大同時使用量を示すグラフと、アプリケーションごとの使用量を示す表が表示されます。[アプリケーションごとの使用量] の表では、特定のアプリケーションについての詳細や、そのアプリケーションを使用しているユーザー、および使用していたユーザーの情報を表示できます。                                                  |
| 仮想マシン使用量の確認                    | [マシン使用量] タブで [デスクトップOSマシン] または [サーバーOSマシン] を選択して、仮想マシンの使用状況をリアルタイムで表示させ、サイトのキャパシティニーズに素早く対処することができます。<br>デスクトップOSの可用性 - デスクトップOSマシン (VDI) の現在の状態をサイト全体または特定のデリバリーグループについて可用性に基づいて表示します。<br><br>サーバーOSの可用性 - サーバーOSマシンの現在の状態をサイト全体または特定のデリバリーグループについて可用性に基づいて表示します。 |
| HDX Insightを使用したネットワーク分析データの表示 | この機能は、組織のライセンスおよび管理者権限によっては使用できない場合があります。<br>[ネットワーク] タブで、ネットワークのユーザー、アプリケーション、およびデスクトップコンテキストビューを表示してネットワーク分析を行います。この機能を使用すると、DirectorでICAトラフィックを高度に分析できます。                                                                                                       |

また、重要なイベントやアクションの発生は、フラグアイコンで示されます。フラグをクリックすると、発生したイベントまたはアクションが表示されます。

注：XenDesktop 7 よりも前のバージョンのVDAについてはHDX接続のログオンデータが収集されません。以前のバージョンのVDAについては、チャートデータが0として表示されます。

以下に示すオプションを使って問題のトラブルシューティングを行います。

- ユーザーマシン電源の制御
- マシンへの接続の無効化

[傾向] ビューでは、各サイトのセッション、接続エラー、マシン障害、ログオンパフォーマンス、および負荷評価についての履歴傾向情報が表示されます。これらの情報を表示するには、[ダッシュボード] または [フィルター] ビューで [傾向] をクリックします。

各グラフのデフォルトの表示範囲を変更するには、[期間] フィルターを変更して適用します。

| 操作 | 説明 |
|----|----|
|    |    |

|                                                |                                                                                                                                                                                                                   |
|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| グラフデータの<br>操作の<br>エクス<br>ポート                   | エクスポートするデータを含むタブを選択します。[エクスポート] をクリックして、エクスポート形式 (PDFまたはCSV) を選択します。                                                                                                                                              |
| セッション<br>の傾向の表<br>示                            | [セッション] タブから、同時接続セッション数に関するより詳細な情報を表示するデリバリーグループと期間を選択します。                                                                                                                                                        |
| 接続エラー<br>の傾向の表<br>示                            | [接続エラー] タブで、接続エラー情報を表示するマシンの種類、エラーの種類、デリバリーグループ、および期間を選択します。                                                                                                                                                      |
| マシン障害<br>の傾向の表<br>示                            | [失敗したデスクトップOSマシン] タブまたは [失敗したサーバーOSマシン] タブで、障害情報を表示するエラーの種類、デリバリーグループ、および期間を選択します。                                                                                                                                |
| ログオンパ<br>フォーマンス<br>の傾向の<br>表示                  | [ログオンパフォーマンス] タブで、ログオンパフォーマンス情報 (ログオン処理時間およびログオン数) を表示するデリバリーグループ、および期間を選択します。このビューには、仲介処理時間や仮想マシンの起動時間などのログオンフェーズにおける平均時間も表示されます。<br>このデータはユーザーのログオンに関するものであり、切断セッションへの再接続は含まれません。                               |
| 負荷評価の<br>傾向の表示                                 | [負荷評価基準インデックス] タブで、サーバーOSマシン間で分散された負荷に関する情報を表示します。このグラフでは、対象のデリバリーグループ、サーバーOSマシン、および期間を指定できます。                                                                                                                    |
| ホストされ<br>たアプリ<br>ケーション<br>の使用量の<br>表示          | この機能は、組織のライセンスによっては使用できない場合があります。<br>[ホストされたアプリケーションの使用量] タブでデリバリーグループと期間を選択すると、最大同時使用量を示すグラフと、アプリケーションごとの使用量を示す表が表示されます。[アプリケーションごとの使用量] の表では、特定のアプリケーションについての詳細や、そのアプリケーションを使用しているユーザー、および使用していたユーザーの情報を表示できます。 |
| HDX Insight<br>を使用した<br>ネットワー<br>ク分析デー<br>タの表示 | この機能は、組織のライセンスおよび管理者権限によっては使用できない場合があります。<br>[ネットワーク] タブで、ネットワークのユーザー、アプリケーション、およびデスクトップコンテキストビューを表示してネットワーク分析を行います。この機能を使用すると、DirectorでICAトラフィックを高度に分析できます。                                                      |

また、重要なイベントやアクションの発生は、フラグアイコンで示されます。フラグをクリックすると、発生したイベントまたはアクションが表示されます。

注: XenDesktop 7 よりも前のバージョンのVDAについてはHDX接続のログオンデータが収集されません。以前のバージョンのVDAについては、チャートデータが0として表示されます。

以下に示すオプションを使って問題のトラブルシューティングを行います。

- ユーザーマシン電源の制御
- マシンへの接続の無効化

特定のマシンのVDA (物理または仮想) にインストールされているHotfixを確認するには、[マシンの詳細] ビューを選択し

ます。

Directorで選択したマシンの電源の状態を制御するには、[電源制御] オプションを使用します。これらのオプションはデスクトップOSマシンに対してのみ実行でき、サーバーOSマシンに対しては使用できません。

注：この機能は、物理マシンまたはリモートPCアクセスを使用しているマシンに対しては使用できません。

| コマンド      | 機能                                                                                                                               |
|-----------|----------------------------------------------------------------------------------------------------------------------------------|
| 再起動       | 仮想マシン上のすべてのプロセスを停止して、通常の再起動処理（ソフト再起動）を実行します。たとえば、起動に失敗したマシンを再起動するときにこのコマンドを使用します。                                                |
| 強制再起動     | 通常のシャットダウン処理を行わずに強制的に仮想マシンを再起動します。これは、物理サーバーの電源プラグを抜いてから電源を入れるのと同等の操作です。                                                         |
| シャットダウン   | 仮想マシン上のすべてのプロセスを停止して、通常のシャットダウン処理（ソフトシャットダウン）を実行します。                                                                             |
| 強制シャットダウン | 通常のシャットダウン処理を行わずに強制的に仮想マシンをシャットダウンします。物理サーバーの電源プラグを抜くのと同等の操作です。実行中のプロセスを正しく停止できない場合があるため、この方法で仮想マシンをシャットダウンするとデータが失われる可能性があります。  |
| サスペンド     | 仮想マシンを一時停止して、そのときの状態をデフォルトのストレージリポジトリ上にファイルとして保存します。この方法で仮想マシンを一時停止してからそのホストサーバーをシャットダウンし、ホストサーバーを再起動してから仮想マシンを元の実行状態に戻すことができます。 |
| 再開        | 一時停止状態の仮想マシンを再開して、元の実行状態に戻します。                                                                                                   |
| 開始        | シャットダウン状態の仮想マシンを起動します（「コールドスタート」とも呼ばれます）。                                                                                        |

電源制御操作に失敗した場合、アラート上にマウスポインターを置くと問題の詳細情報がポップアップメッセージとして表示されます。

メンテナンスモードでは、管理者がイメージの保守作業を行っている間、一時的にユーザーが接続できなくなります。

マシンをメンテナンスモードにすると、メンテナンスモードを解除するまでそのマシンへの接続が禁止されます。そのマシンにユーザーがログオンしている場合は、すべてのユーザーがログオフした後でメンテナンスモードに切り替わります。ユーザーのログオフを促すには、マシンのシャットダウンを通知するメッセージをユーザーに送信したり、電源制御機能を使って強制的にマシンをシャットダウンしたりできます。

1. [ユーザーの詳細] ビューなどからマシンを選択するか、[フィルター] ビューでマシンのグループを選択します。
2. [メンテナンスモード] をクリックしてマシンをメンテナンスモードにします。

メンテナンスモードのデスクトップにユーザーが接続を試みると、デスクトップを使用できないことを示すメッセージが表示

されます。管理者がメンテナンスモードを解除するまで、新しい接続は許可されません。

# ユーザーの問題のトラブルシューティング

Jan 04, 2017

Directorの [ヘルプデスク] ビュー ( [アクティビティマネージャー] ページ) を使って、ユーザーに関する情報を確認します。

- ユーザーのログオン、接続、およびアプリケーションの状態について確認する。
- ユーザーのマシンをシャドウする。
- 次の表に示す方法で問題のトラブルシューティングを行い、必要な場合は問題を担当の管理者に報告する。

## トラブルシューティングのヒント

| ユーザーの問題                     | 参照先                                 |
|-----------------------------|-------------------------------------|
| ログオンに時間がかかる。断続的もしくは繰り返し失敗する | <a href="#">ユーザーログオンの問題の診断</a>      |
| アプリケーションが遅いまたは応答しない         | <a href="#">アプリケーションエラーの解決</a>      |
| 接続できない                      | <a href="#">デスクトップ接続の復元</a>         |
| セッションが遅いまたは応答しない            | <a href="#">セッションの復元</a>            |
| ビデオが遅いまたは画質が悪い              | <a href="#">HDXチャンネルシステムレポートの生成</a> |

注： [ユーザーの詳細] ビューの [マシンの詳細] パネルで、マシンがメンテナンスモードになっていないことを確認してください。

## 検索のヒント

Directorの [検索] フィールドにユーザー名を入力すると、Directorのサポートが構成されたすべてのサイトでActive Directoryユーザーが検索されます。

[検索] フィールドにマルチユーザーマシンの名前を入力すると、そのマシンの [マシンの詳細] ページが開きます。

[検索] フィールドにエンドポイントの名前を入力すると、そのエンドポイントに接続している認証が不要なユーザー (匿名ユーザー) セッションおよび認証が必要なセッションを検索でき、匿名ユーザーセッションのトラブルシューティングを行うことができます。匿名ユーザーセッションのトラブルシューティングを行うには、エンドポイント名が重複していないことが重要です。

検索結果には、現在マシンを使用していないユーザーや、マシンに割り当てられていないユーザーも含まれます。

- 検索では大文字と小文字は区別されません。
- 検索語の一部を入力すると、一致する候補が一覧で表示されます。
- ユーザー名、姓と名、または表示名などをスペースで区切って複数の文字列として入力すると、両方の文字列と一致する項目が検索されます。たとえば、jo robと入力すると、「John Robertson」や「Robert, Jones」などが検索されます。

ホームページに戻るには、Directorのロゴをクリックします。

## Citrixテクニカルサポートへのトラブルシューティング情報のアップロード

単一のDelivery ControllerまたはVirtual Delivery AgentからCitrix Scoutを実行し、選択したコンピューターのトラブルシューティングに必要なデータ要素やCitrix Diagnosis Facility (CDF) トレースをキャプチャします。キャプチャされた情報は、Scoutにより安全にCitrixテクニカルサポートにアップロードされます。TaaS (Tools As a Service) プラットフォームでは、これらの情報を使用して、報告された問題を短時間で解決します。

Scoutは、XenAppまたはXenDesktopのコンポーネントと一緒にインストールされます。XenDesktop 7.1、XenDesktop 7.5、またはXenApp 7.5をインストールすると、Windowsのスタートメニューやスタート画面にScoutが追加されます。

スタートメニューやスタート画面からScoutを起動するには、[Citrix]、[Citrix Scout] の順に選択します。

Scoutの使用と構成、および一般的な問題について詳しくは、<http://support.citrix.com/article/CTX134119>を参照してください

Scoutの使用方法については、次のビデオを参照してください。

# ユーザーのシャドウ

Jan 04, 2017

Directorのユーザーのシャドウ機能を使用すると、ユーザーの仮想デスクトップまたはセッションを直接表示したり操作したりできます。この機能を使用するには、そのマシンにユーザーが接続している必要があります。ユーザーが接続している場合、ユーザーのタイトルバーにそのマシン名が表示されます。

1. [ユーザーの詳細] ビューで、ユーザーセッションを選択します。
2. 以下の操作で、そのユーザーセッションに対するシャドウを開始します。
  - マシンを監視する場合は、[アクティビティマネージャー] ビューで [シャドウ] をクリックします。
  - セッションを監視する場合は、[ユーザーの詳細] ビューの [セッション詳細] パネルで [シャドウ] をクリックします。
3. 接続が初期化されると、.msrcincidentファイルを開くか保存するかを確認するダイアログボックスが開きます。
4. デフォルトで選択されていない場合は、Remote Assistance Viewerでファイルを開きます。ユーザーデバイス側には、確認のメッセージが表示されます。
5. ユーザーが [はい] をクリックすると、マシンまたはセッションの共有が開始されます。

ユーザーがマウスやキーボードの制御を許可すると、管理者がシャドウセッションを制御できるようになります。

Microsoft Internet ExplorerブラウザーでダウンロードしたMicrosoftリモートアシスタンスファイル (.msra) がリモートアシスタンスクライアントで自動的に開くように構成します。

これを行うには、グループポリシーエディターで [ファイルのダウンロード時に自動的にダイアログを表示] を有効にする必要があります。

[コンピューターの構成] > [管理用テンプレート] > [Windowsコンポーネント] > [Internet Explorer] > [インターネットコントロールパネル] > [セキュリティページ] > [インターネットゾーン] > [ファイルのダウンロード時に自動的にダイアログを表示]

デフォルトでは、ローカルイントラネットゾーンのサイトに対してこのオプションが有効になっています。Directorサイトがローカルイントラネットゾーンにない場合は、ローカルイントラネットゾーンに追加することを検討してください。

# ユーザーへのメッセージの送信

Jan 04, 2017

Directorでは、マシンに接続しているユーザーにメッセージを送信できます。たとえば、突発的にデスクトップの保守、ログオフ、再起動、プロファイルのリセットなどが必要になった場合に、ユーザーに緊急のメッセージを送信できます。

1. [アクティビティマネージャー] ビューでユーザーを選択して、[詳細] をクリックします。
2. [ユーザーの詳細] ビューの [セッション詳細] パネルで、[メッセージの送信] をクリックします。
3. 送信するメッセージの [件名] および [メッセージ] を入力して、[送信] をクリックします。

メッセージが正しく送信されると、Directorに確認メッセージが表示されます。マシンに接続しているユーザーにメッセージが表示されます。

メッセージの送信に問題が発生すると、Directorにエラーメッセージが表示されます。そのエラーメッセージに従って問題を解決してください。問題を解決したら、件名およびメッセージを再入力して [再試行] をクリックします。



# ユーザーログオンの問題の診断

Jan 04, 2017

以下の手順で診断を行います。

1. ログオン状態のトラブルシューティングを行うには、[ユーザーの詳細] ビューの [ログオン処理時間] パネルを使用します。
  - ユーザーがログオン中の場合は、ここにログオンのプロセスが表示されます。
  - ユーザーがログオン済みの場合、ユーザーがそのセッションにログオンするときにかかった時間が [ログオン処理時間] パネルに表示されます。
2. ユーザーにいったんログオフしてから再度ログオンしてもらい、[ログオン処理時間] パネルに表示される情報を確認します。通常、パネルの内容は約3分後に更新されますが、ログオン処理にかかる時間により長くなることがあります。
3. ログオン時の各フェーズを調査します。
  - **仲介** — ユーザーに割り当てるデスクトップを決定するのに要した時間。
  - **VM起動** — デスクトップの起動に要した時間。
  - **HDX接続** — HDX接続の確立に要した時間（ネットワークに依存）。
  - **GPO** — グループポリシーオブジェクトの適用に要した時間。
  - **ログオンスクリプト** — スクリプトに要した時間。
  - **プロファイルロード** — ユーザープロファイルのロードに要した時間。
  - **対話型セッション** — 対話型ユーザーセッションの確立に要した時間。

総ログオン時間は、これらの各フェーズを厳密に合計したものではありません。たとえば、複数のフェーズが並行して発生したり、一部のフェーズで追加処理が発生したりしてログオン処理時間が合計値よりも大きくなる場合があります。

ヒント：グラフで異常または予期しない値を識別するには、現在のセッションの各フェーズで要した時間と、このユーザーの最近7日間の平均処理時間、およびこのデスクトップグループのすべてのユーザーの最近7日間の平均処理時間を比較します。

必要に応じて、担当管理者に報告します。たとえば、仮想マシンの起動に時間がかかり、ハイパーバイザーが問題の原因である可能性がある場合は、ハイパーバイザー管理者に問題を報告します。また、仲介処理に時間がかかる場合は、サイト管理者に Delivery Controller の負荷分散のチェックを依頼します。

トラブルシューティングのヒント：以下の問題について調査します。

- (現在の) ログオンを示すバーが表示されていない。
- 現在のログオン処理時間とこのユーザーの平均処理時間が大きく食い違う。次の原因が考えられます。
  - 新しいアプリケーションがインストールされた。
  - オペレーティングシステムが更新された。
  - 構成が変更された。
- ユーザーのログオン処理時間（現在値および平均値）とデリバリーグループの平均値が大きく食い違う。

必要な場合は、[再起動] をクリックしてユーザーに再ログオンしてもらい、仮想マシンの起動や仲介時に問題が発生するかどうかを確認します。

# アプリケーションエラーの解決

Jan 04, 2017

[アクティビティマネージャー] ビューで [アプリケーション] タブをクリックします。ここでは、このユーザーがアクセスするすべてのマシン上のすべてのアプリケーションとその状態を確認できます。これには、現在接続しているマシンのローカルアプリケーションおよびホストされるアプリケーションが含まれます。

注： [アプリケーション] タブが灰色表示になっている場合は、このタブを有効にする権限を持つ管理者に問い合わせてください。

一覧には、セッション内で起動されたアプリケーションのみが表示されます。

サーバーOSマシンおよびデスクトップOSマシンでは、アプリケーションが切断セッションごとに一覧で表示されます。ユーザーが接続していない場合、アプリケーションは表示されません。

| 操作                   | 説明                                                                                                                                                                                                      |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 応答していないアプリケーションを終了する | 応答していないアプリケーションを選択し、[アプリケーションの終了] をクリックします。アプリケーションが終了したら、ユーザーに再度起動するように通知します。                                                                                                                          |
| 応答していないプロセスを終了する     | 必要な権限がある場合は、[プロセス] タブをクリックします。アプリケーションに関連するプロセス、またはCPUリソースやメモリを過度に消費しているプロセスを選択し、[プロセスの終了] をクリックします。<br>プロセスを終了するための権限がない場合、プロセスを終了することはできません。                                                          |
| ユーザーのマシンを再起動する       | デスクトップOSマシンでは、選択したセッションで[再起動] をクリックします。<br>または、[マシンの詳細] ビューで電源制御を使ってマシンを再起動またはシャットダウンします。アプリケーションの状態を再確認するには、ユーザーに再度ログオンするように通知します。<br><br>サーバーOSマシンでは、[再起動] オプションを使用できません。代わりに、ユーザーをログオフして、再度ログオンさせます。 |
| マシンをメンテナンスモードにする     | パッチまたはそのほかの更新などによりマシンのイメージをメンテナンスする必要がある場合は、マシンをメンテナンスモードにして、担当の管理者に問題を報告します。[マシンの詳細] ビューで [詳細] をクリックして、メンテナンスモードのオプションをオンにします。担当の管理者に報告します。                                                            |

# デスクトップ接続の復元

Jan 04, 2017

Directorビューでは、タイトルバーにそのユーザーの接続状態が表示されます。

デスクトップ接続に問題が発生するとその原因が表示されるため、トラブルシューティング方法を判別することができます。

| 操作                      | 説明                                                                                       |
|-------------------------|------------------------------------------------------------------------------------------|
| マシンがメンテナンスモードでないことを確認する | [ユーザーの詳細] ページで、メンテナンスモードがオフであることを確認します。                                                  |
| ユーザーのマシンを再起動する          | マシンを選択し、[再起動] をクリックします。ユーザーのマシンがCPUリソースを過度に消費しているためにマシンが応答しないまたは接続できない場合は、このオプションを使用します。 |

# セッションの復元

Jan 04, 2017

セッションが切断状態になると、セッションおよびアプリケーションは終了しませんが、サーバーとユーザーデバイス間の通信が切断されます。

[ユーザーの詳細] ビューで、[セッション詳細] パネルのセッション障害のトラブルシューティングを実行します。現在のセッションがセッションIDで示され、詳細を確認できます。

| 操作                          | 説明                                                                                                                                              |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| 応答していないアプリケーションまたはプロセスを終了する | [アプリケーション] タブをクリックします。応答していないアプリケーションを選択し、[アプリケーションの終了] をクリックします。同様に、応答していないプロセスを選択し、[プロセスの終了] をクリックします。<br>また、メモリやCPUリソースを過度に消費しているプロセスを終了します。 |
| Windowsセッションを切断する           | [セッション制御] をクリックし、[切断] を選択します。このオプションは、仲介されたサーバーOSマシンに対してのみ使用できます。仲介されていないセッションでは無効です。                                                           |
| セッションからユーザーをログオフする          | [セッション制御] をクリックし、[ログオフ] を選択します。                                                                                                                 |

セッション障害が解決されたことを確認するために、ユーザーに再度ログオンさせます。また、ユーザーをシャドウしてセッションをより詳しく監視することもできます。

注：ユーザーデバイスでXenDesktop 7より前のバージョンのVirtual Delivery Agent (VDA) が動作している場合、Directorはセッションに関する完全な情報を表示することができません。代わりに、情報を使用できないことを示すメッセージが表示されます。このメッセージは、[ユーザーの詳細] ページおよび[アクティビティマネージャー] に表示されます。

# HDXチャネルシステムレポートの生成

Jan 04, 2017

ユーザーのマシン上のHDXチャネルの状態を確認するには、[ユーザーの詳細] ビューの [HDX] パネルを使用します。このパネルは、HDXを使ってユーザーマシンに接続している場合にのみ操作できます。

情報を使用できないことを示すメッセージが表示された場合は、ページが更新されるまで1分待つか、[更新] ボタンをクリックしてください。HDXデータはほかのデータより更新に時間がかかることがあります。

エラーまたは警告のアイコンをクリックすると、詳細が表示されます。

ヒント：このダイアログボックスでは、タイトルバーの左隅にある矢印をクリックしてほかのチャネルの情報を表示することもできます。

HDXチャネルシステムレポートは、主にCitrixサポートチームによるトラブルシューティング時に使用されます。

1. [HDX] パネルで、[システムレポートのダウンロード] をクリックします。
2. 生成されたXML形式のレポートファイルを表示したり保存したりできます。
  - XMLファイルを表示するには、[開く] をクリックします。DirectorにXMLファイルの内容が表示されます。
  - XMLファイルを保存するには、[保存] をクリックします。[名前を付けて保存] ダイアログボックスで、ファイルの保存場所としてDirectorが動作するマシン上のフォルダーを指定します。

# Personal vDiskのリセット

Jan 04, 2017

注意：Personal vDiskをリセットすると、設定がデフォルトの状態にリセットされ、そのディスク上のすべてのデータが削除されます。ただし、Personal vDiskのデフォルト設定（Cドライブからのプロファイルリダイレクトの設定）を変更しない限り、またはサードパーティ製プロファイル管理ソリューションを使用していない場合、Personal vDiskをリセットしてもプロファイルデータは保持されます。

Personal vDiskをリセットするには、そのPersonal vDiskを使用しているマシンが実行中である必要があります。ユーザーがログオンしていなくても構いません。

このオプションはデスクトップOSマシンに対してのみ使用できます。サーバーOSマシンでは無効です。

1. [ヘルプデスク] ビューでデスクトップOSマシンを選択します。
2. このビュー、または[ユーザーの詳細] ビューの[個人設定] パネルで、[Personal vDiskのリセット] をクリックします。
3. [リセット] をクリックします。ユーザーがログオフされることを警告するメッセージが表示されます。ログオンしていたユーザーはログオフされ、マシンが再起動します。

リセットに成功すると、[ユーザーの詳細] ビューの[個人設定] パネルの[Personal vDiskの状態] に[実行中] と表示されます。リセットに失敗すると、[実行中] の右側に赤いXが表示されます。このX上にマウスポインターを置くと、問題についての情報が表示されます。

# ユーザープロファイルのリセット

Jan 04, 2017

注意：プロファイルのリセットすると、そのユーザーのフォルダーやファイルは保存され、新しいプロファイルにコピーされます。ただし、多くのユーザープロファイルデータは削除されます。たとえば、レジストリはリセットされ、アプリケーション設定も削除される場合があります。

1. Directorからリセットするプロファイルを持つユーザーを検索し、そのユーザーのセッションを選択します。
2. [プロファイルのリセット] をクリックします。
3. ユーザーに、すべてのセッションからログオフするように指示します。
4. ユーザーにログオンするように指示します。ユーザープロファイルから保存されたフォルダーやファイルが新しいプロファイルにコピーされます。

重要：複数のプラットフォーム上（Windows 8とWindows 7など）にユーザーのプロファイルが存在する場合は、問題が発生したデスクトップまたはアプリケーションに最初にログオンするよう指示します。これにより、正しいプロファイルがリセットされます。

Citrixユーザープロファイルの場合、ユーザーのデスクトップが表示された時点でリセットされています。Microsoftの移動プロファイルの場合、フォルダーの復元処理に時間がかかる場合があります。この復元処理が完了するまで、ユーザーはログオンしていません。

注：これまでの手順では、XenDesktop（デスクトップVDA）を使用している前提になっています。XenApp（サーバーVDA）を使用している場合は、プロファイルのリセットを実行するためにログオンする必要があります。ユーザーはいったんログオフしてから再度ログオンし、プロファイルのリセットを完了させる必要があります。

プロファイルが正しくリセットされない場合（ユーザーがそのマシンに再ログオンできなかつたり一部のファイルが見つからなかつたりする場合など）、管理者が手作業で元のプロファイルを復元する必要があります。

ユーザーのプロファイルのフォルダーやファイルが保存され、新しいプロファイルにコピーされます。これらのフォルダーは、以下の順番でコピーされます。

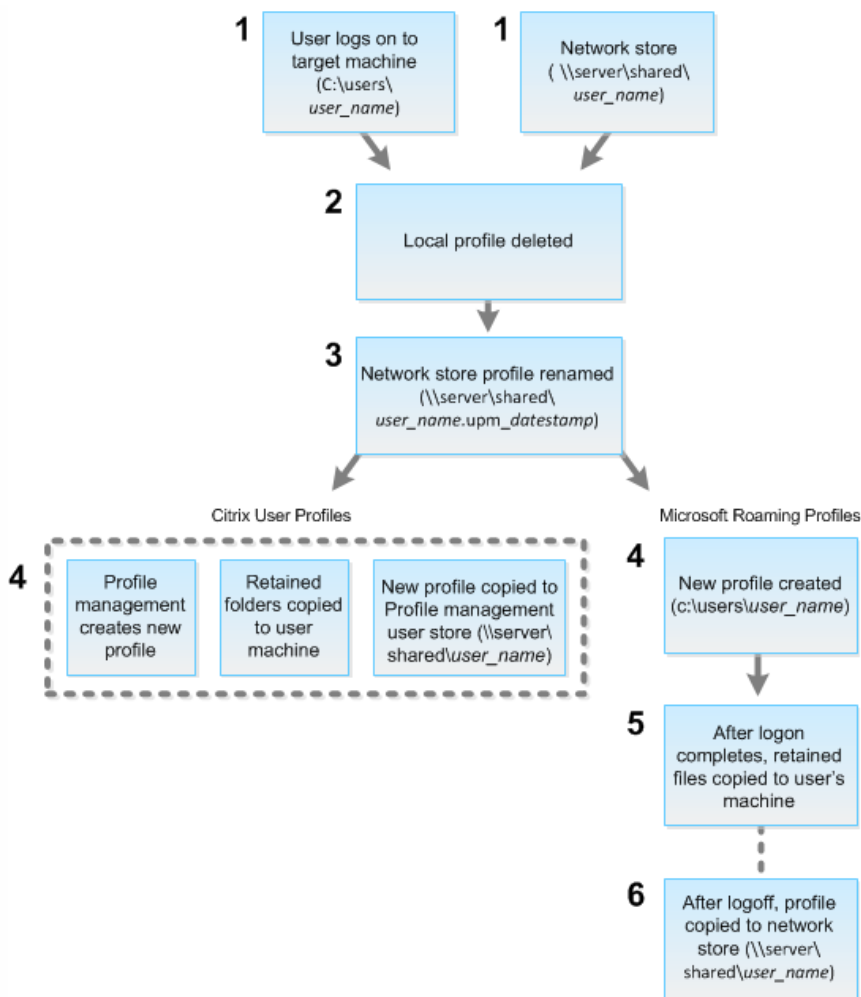
- デスクトップ
- クッキー
- お気に入り
- ドキュメント
- マイピクチャ
- マイミュージック
- マイビデオ

注：Windows 8以降では、プロファイルのリセット時にクッキーフォルダーはコピーされません。

リセットされたプロファイルはどのように処理されるか

いずれのCitrixユーザープロファイルまたはMicrosoft移動プロファイルもリセットできます。ユーザーがログオフした後に管理者がDirectorまたはPowerShell SDKでリセットコマンドを選択すると、使用されているユーザープロファイルが識別され、Directorにより適切なリセットコマンドが発行されます。DirectorはProfile managementを介してプロファイルのサイズ、種類、およびログオン時間などに関する情報を取得します。

次にユーザーがログオンしたときに、下図の処理が実行されます。



1. Directorからのリセットコマンドにより、プロファイルの種類が指定されます。次に、Profile managementサービスによりその種類のプロファイルのリセットが試行され、適切なネットワーク共有（ユーザーストア）が検出されます。Profile managementにより処理されたユーザーのプロファイルに対して移動プロファイル用のコマンドが発行された場合は拒否されます（逆の場合も同様）。
2. ローカルプロファイルがある場合は削除されます。
3. ネットワークプロファイルの名前が変更されます。
4. 次の処理は、リセットされるプロファイルがCitrixユーザープロファイルかMicrosoft移動プロファイルかにより異なります。
  - Citrixユーザープロファイルの場合、Profile managementのインポート規則によって新しいプロファイルが作成され、フォルダーがネットワークプロファイルにコピーされ、ユーザーのログオンが通常どおりに処理されます。リセットに移動プロファイルが使用される場合は、移動プロファイル内のすべてのレジストリ設定がリセットプロファイル内に保持されます。  
注：必要な場合は、テンプレートプロファイルが移動プロファイルよりも優先されるようにProfile managementを構成することもできます。
  - Microsoft移動プロファイルの場合、Windowsにより新しいプロファイルが作成され、ユーザーがログオンするとフォルダーがユーザーデバイスにコピーされます。ユーザーが再度ログオフすると、新しいプロファイルがネットワークストアにコピーされます。

リセットに失敗したプロファイルを手動で復元するには

1. ユーザーに、すべてのセッションからログオフするように指示します。
2. ローカルプロファイルが存在する場合はそれを削除します。



3. ネットワーク共有上のアーカイブフォルダーを検索します。アーカイブフォルダーには、拡張子として日時が含まれます (.upm\_<datestamp>)。
4. 現在のプロファイルのフォルダー（拡張子upm\_<datestamp>のないもの）を削除します。
5. アーカイブフォルダーの名前から日時拡張子を削除します。プロファイルがリセット前の状態に戻りました。

# Session Recording - XenApp 7.6 FP1、FP2、および LTSR

Jan 04, 2017

Session Recordingを使用すると、接続の種類を問わず、企業ポリシーおよび法規制に従って、サーバーOSのVDAマシンからホストされる任意のユーザーセッションの、画面上のアクティビティを録画することができます。Session Recordingにより録画、カタログ化、およびアーカイブされるセッションを、後で取得して再生することができます。

Session Recordingでは柔軟性の高いポリシーを使用して、アプリケーションセッションの録画を自動的に起動します。IT部門ではSession Recordingを使用して、財務業務や病院での患者情報のシステムなどでエンドユーザーがアプリケーションをどのように使用するかを調査できます。内部統制をサポートすることによって、法規制の順守を徹底し、セキュリティ監視を成功に導きます。同様に、技術サポート部門でこの機能を使用すれば、問題の特定と解決までの時間を短縮することができます。

## 長所

**記録と監視によるセキュリティの向上。** Session Recordingにより、機密情報を取り扱うアプリケーションで、エンドユーザーの画面上での操作を録画できます。これは、医療や金融などの規制の厳しい業界では、特に重要な機能です。このような業界では、記録が禁止されている個人情報を取り扱っているため、ポリシーによって選択的な録画を実行できます。

**強力なアクティビティ監視。** 画面の更新、マウスのクリック、および目に見えるキーボード入力をデジタル署名されたビデオに録画および保存し、特定のエンドユーザー、アプリケーション、およびサーバーの操作を録画できます。

Session Recordingは、法的手続きの証拠収集用に設計、開発されてはいません。Session Recordingを使用している組織が証拠の収集を行う場合は、一般的な動画録画とテキストベースの電子証拠開示ツールの組み合わせなどを使用することをお勧めします。

**迅速な問題解決。** 再現が難しい問題についてエンドユーザーが問い合わせをしたときに、ヘルプデスクのサポートスタッフがユーザーセッションを録画できます。問題が再発したら、発生時刻が記録されているエラーの録画を使用して、より迅速に問題のトラブルシューティングができます。

# Session Recordingの導入

Jan 04, 2017

次の作業を行うと、XenAppセッションを録画し、検査できるようになります。

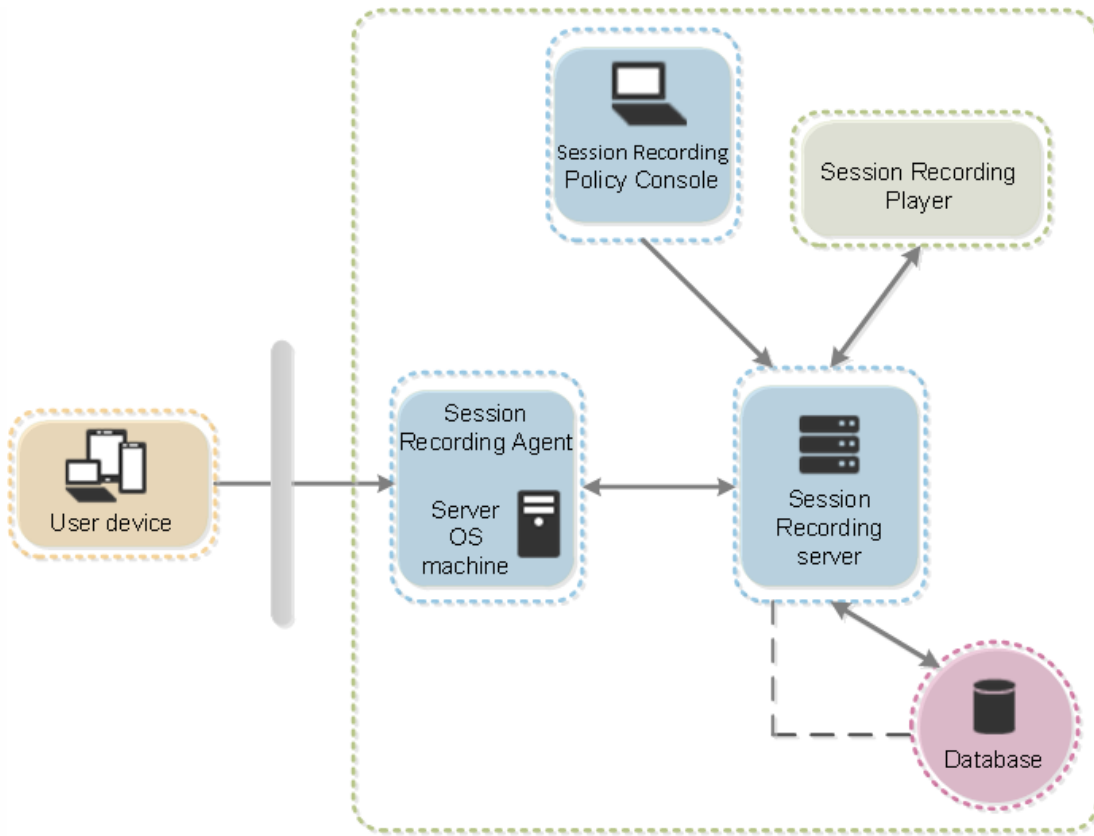
1. Session Recordingコンポーネントを理解する。
2. 環境に合う展開シナリオを選択する。
3. インストール要件を確認する。
4. Session Recordingをインストールする。
5. Session Recordingコンポーネントを構成して、録画およびセッションの表示を許可する。

Session Recordingは5つのコンポーネントから構成されます。

- **Session Recording Agent**。各サーバーOSマシンにインストールする、録画処理を有効にするコンポーネントです。これによりセッションデータが録画されます。
- **Session Recording Server**。次のサービスをホストするサーバーです。
  - **ブローカー**：IIS 6.0以降によりホストされるWebアプリケーションです。これにより、Session Recording Playerからの検索クエリおよびファイルダウンロード要求と、Session Recording Policy Consoleからのポリシー管理要求が制御されます。また、XenAppの各セッションの録画ポリシーが評価されます。
  - **ストレージマネージャー**：Windowsサービスです。これにより、XenAppを実行中のSession Recordingが有効な各コンピューターから受信する、セッションの録画ファイルが管理されます。
- **Session Recording Player**。XenAppセッションのファイルを調査するユーザーが、録画を再生するためにワークステーションでアクセスするユーザーインターフェイスです。
- **Session Recordingデータベース**。セッションの録画データを格納するためのSQLデータベースです。
- **Session Recording Policy Console**。録画するセッションを指定するポリシーを作成するコンソールです。

この図はSession Recordingコンポーネントおよび各コンポーネントの関係を示しています。

この展開例では、Session Recording Agent、Session Recording Server、Session Recordingデータベース、Session Recording Policy Console、およびSession Recording Playerのすべてが、セキュリティファイアウォールの内部に設置されています。Session Recording AgentはサーバーOSマシンにインストールされます。第2のサーバーはSession Recording Policy Consoleをホストし、第3のサーバーはSession Recording Serverとして機能します。そして、第4のサーバーはSession Recordingデータベースをホストします。Session Recording Playerはワークステーションにインストールされます。ファイアウォール外音のクライアントデバイスはSession Recording AgentがインストールされているサーバーOSマシンに接続します。ファイアウォール内では、Session Recording Agent、Session Recording Policy Console、Session Recording Player、およびSession RecordingデータベースはすべてSession Recording Serverに接続します。



# 導入計画

Jan 04, 2017

環境に応じ、異なるシナリオに基づいてSession Recordingコンポーネントを展開できます。

Session Recordingは、単一のサイトのみに限って展開する必要はありません。Session Recording Agent以外はすべて、サーバーサイトに依存しないコンポーネントです。たとえば、複数のサイトで単一のSession Recording Serverを使用するように設定できます。

複数のエージェントを展開した大規模なサイトがあり、画像処理にかなりリソースを消費するアプリケーション（AutoCADなど）を数多く録画したり、多数のセッションを録画したりすることが予測される場合、Session Recording Serverに非常に負荷がかかる可能性があります。パフォーマンスの問題を解決するために、複数のSession Recording Serverを異なるコンピューターにインストールし、Session Recording Agentがそれらのコンピューターと連携するように設定できます。エージェントで1度に参照できるのは1つのサーバーのみであることに留意してください。

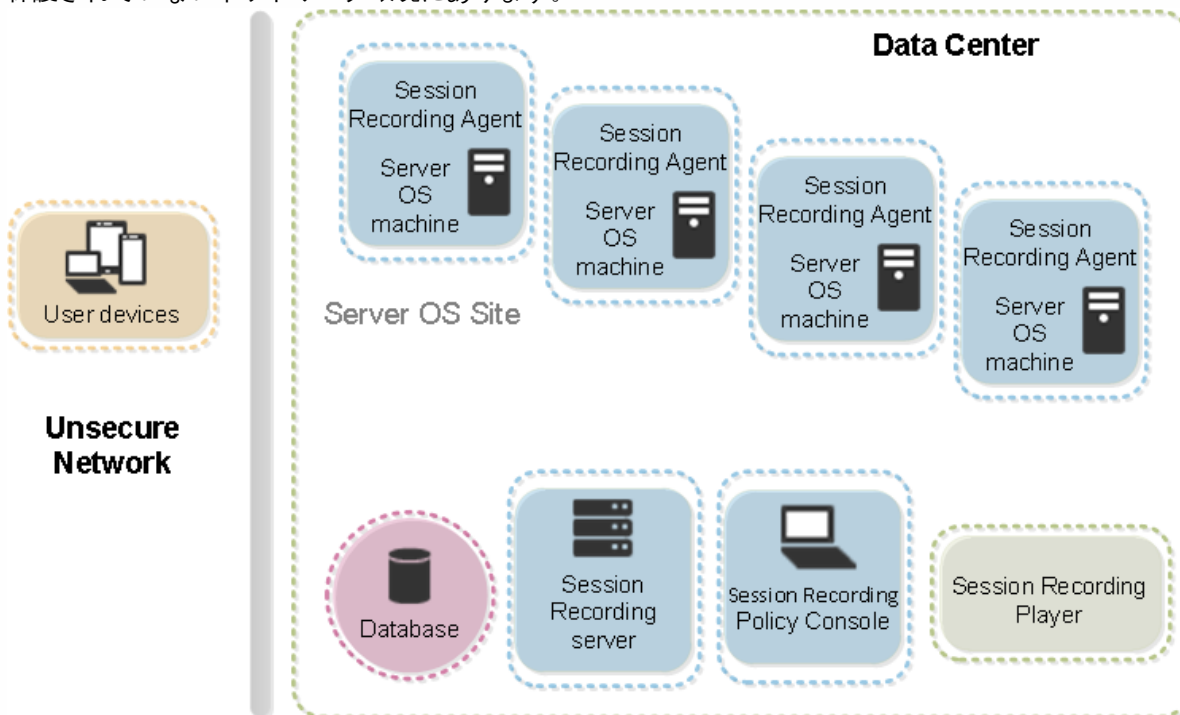
## 展開シナリオの例

ここでは、Session Recordingを展開する場合の2種類の構成例について説明します。

- 単一のサーバーOSマシンにSession Recording Agentを展開する。
- サイトの複数のサーバーOSマシンにSession Recording Agentを展開する。

## サーバーサイトの構成

1つまたは複数のサイトのセッションを録画する場合は、この構成を使用します。Session Recording Agentはサイト内の各サーバーOSマシンにインストールされます。サイトはセキュリティファイアウォール内のデータセンターにあります。Session Recording Administrationコンポーネント（Session Recordingデータベース、Session Recording Server、Session Recording Policy Console）を別の複数のサーバーにインストールし、Session Recording Playerはワークステーションにインストールします。これらのコンピューターはファイアウォール内ですがデータセンターの外部にあります。ワークステーション、モバイルデバイス、およびラップトップコンピューターのようなXenAppクライアントは、ファイアウォール外部の保護されていないネットワーク環境にあります。





# セキュリティの推奨事項

Jan 19, 2017

Session Recordingは、安全なネットワーク上に展開され管理者によりアクセスされるものとして設計されており、そのこと前提にセキュリティを維持するコンポーネントです。設計上、デフォルトの構成はシンプルなシステムです。デジタル署名や暗号化などのセキュリティ機能はオプションで設定できます。

Session Recordingコンポーネント間の通信は、インターネットインフォメーションサービス (IIS) とMicrosoftメッセージキュー (MSMQ) を通じて実現されます。IISにより、各Session Recordingコンポーネント間のWebサービスの通信リンクが提供されます。MSMQにより、Session Recording AgentからSession Recording Serverへセッションの録画データを送信するための、信頼できるデータ伝送メカニズムが提供されます。

展開計画を立てるときに、セキュリティに関する次の推奨事項について検討します。

- Session Recordingコンポーネントが動作するサーバーを物理的に保護する。可能であれば、権限を持つ人のみが入室できる安全なサーバー室にコンピューターを設置します。
- Session Recordingコンポーネントが動作するサーバーを別のサブネットまたはドメインに分離する。
- Session Recording Serverとほかのサーバーの間にファイアウォールを設置し、ほかのサーバーにアクセスするユーザーからセッションの録画データを保護する。
- Microsoftからの最新のセキュリティアップデートにより、Session Recording Admin ServerおよびSQLデータベースを最新に保ちます。
- 管理者以外の方が管理マシンにログオンできないように制限する。
- 録画ポリシーの変更およびセッションの録画ファイルの表示を行う権限を持つユーザーを厳しく制限する。
- デジタル証明書をインストールし、Session Recordingのファイル署名機能を使用し、IISでSSL通信をセットアップする。
- [Session Recording Agentのプロパティ] ダイアログボックスに表示されるMSMQプロトコルをHTTPSに設定して、MSMQの通信でHTTPSが使用されるようにする。詳しくは、「[MSMQのトラブルシューティング](#)」を参照してください。
- TLS 1.0を使いSSLv2、SSLv3、およびRC4 暗号をSession Recording ServerとSession Recordingデータベースで無効にします。詳しくは、Microsoftアールクル<http://support.microsoft.com/default.aspx?scid=kb;ja-jp;187498>および<http://support.microsoft.com/kb/245030/en-us>を参照してください。
- 再生データの保護機能を使用する。再生データの保護はSession Recordingの機能の1つで、これにより、Session Recording Playerにダウンロードされる前に、セッションの録画ファイルが暗号化されます。このオプションは [Session Recording Server Properties] にあり、デフォルトで有効に設定されます。
- 暗号化キー長および暗号化アルゴリズムのNSITガイダンスに従います。

Session Recording機能の構成について詳しくは、<http://support.citrix.com/article/CTX200868>を参照してください。

## 証明書のインストール

Session Recording Serverがインストールされているコンピューターでは、Session Recording Agent、Session Recording Player、またはSession Recording Policy ConsoleからSSL接続が行われると、IIS Webサーバーによりそのサーバー証明書がクライアントに送信されます。サーバー証明書を受信すると、Session Recording Agent、Session Recording Player、またはPolicy Consoleでは、証明書を発行したCA (Certificate Authority : 証明機関) と、クライアントによってそのCAが信頼されているかどうか判定されます。CAが信頼されていない場合は証明書を拒否されます。Session Recording Agentの場合はアプリケーションイベントログにエラーが記録され、Session Recording PlayerまたはPolicy Consoleの場合はユーザーにエラーメッセージが表示されます。

サーバー証明書は、サーバーについて情報を収集し、CAにそのサーバーの証明書の発行を要求することによってインストールします。サーバー証明書を要求したり、サーバー名を確認したりするとき、正確な情報を指定する必要があります。クライアント接続 (Session Recording Agent、Session Recording Player、およびPolicy Console) で完全修飾ドメイン名が使用されている場合は、CAに指定する証明書情報でもサーバーのNetBIOS名ではなく完全修飾ドメイン名を使用する必要があります。NetBIOS名を指定する場合は、サーバー証明書を要求するときに完全修飾ドメイン名を指定しないでください。サーバー証明書をローカルサーバーの証明書ストアにインストールします。発行元のCA証明書を各クライアントにインストールします。

Session Recordingで使用できるサーバー証明書を発行するCAが、組織内に構築されている場合があります。内部CAを使用する場合は、発行元のCA証明書を各クライアントデバイスに必ずインストールしてください。証明書とCAの使用方法については、Microsoft社のドキュメントを参照してください。外部CAを使用する場合は、VeriSign、Baltimore、Entrust、およびそれらの関連企業などの企業や組織が、現在CAとして活動しています。

すべての証明書には、CAが定義する失効日があります。失効日は証明書のプロパティで確認することができます。Session Recordingでエラーが発生しないように、失効日前に必ず証明書を更新してください。

Session Recordingをインストールすると、HTTPSを使用するようにデフォルトで設定されます。この場合、CAから発行されたサーバー証明書を使用して既定のWebサイトを設定する必要があります。IISでのサーバー証明書のインストール手順については、IISのドキュメントを参照してください。



# スケーラビリティに関する注意事項

Jan 04, 2017

Session Recordingのインストールと実行のために、XenAppの実行に必要なハードウェア要件を超えて、さらにリソースを追加する必要はほとんどありません。ただし、Session Recordingを使用して数多くのセッションを録画する予定である場合、または画像処理に多くのリソースを消費するアプリケーションを録画するなど、セッションの録画ファイルのサイズが大きくなるのが予測される場合は、Session Recordingの展開計画を立てるときに、使用するシステムのパフォーマンスについて検討します。

高スケーラブルなSession Recordingシステムの構築について詳しくは、<http://support.citrix.com/article/CTX200869>」を参照してください。

ここでは次のことについて説明します。

## ハードウェアの推奨事項

### ディスクとストレージ

### ネットワークの性能

### コンピューターの処理性能

### 複数のSession Recording Serverの展開

### データベースのスケーラビリティ

## ハードウェアの推奨事項

各Session Recording Serverに送信するデータの量と、サーバーでこのデータを処理し格納するのにかかる時間について考えます。受信データを格納する速度がデータ入力速度より高速である必要があります。

データ入力速度を推定するには、録画対象のセッションの数をその平均サイズで乗算してから、セッションの録画期間で除算します。たとえば、5,000件のMicrosoft Outlookのセッションの平均サイズが20MBとして、毎営業日に8時間録画するとします。この場合は、データ入力速度はおよそ3.5MBpsです (5,000セッション×20MB÷8時間÷3,600秒)。

単一のSession Recording Serverのパフォーマンスを最適化するか、複数のコンピューターにSession Recording Serverをインストールすることによって、パフォーマンスを向上させることができます。

## ディスクとストレージ

ディスクとストレージは、Session Recordingの展開を計画するときには考慮すべき最も重要な要素です。使用するストレージソリューションの書き込み性能は特に重要です。データの書き込み速度が速いほど、システム全体のパフォーマンスが向上します。

ローカルディスクコントローラーまたは接続されたSAN (Storage Area Network : ストレージエリアネットワーク) により、RAIDアレイとしてローカルディスクのセットを制御する方法は、Session Recordingでの使用に適したストレージソリューションです。

注 : Session Recordingでは、NAS (Network-Attached Storage : ネットワークアタッチトストレージ) を使用しないでください。ネットワークドライブへの録画データの書き込みに関連して、パフォーマンスおよびセキュリティ上の問題が起きる可能性があります。

ローカルドライブにキャッシュメモリが組み込まれているディスクコントローラーを使用すると、パフォーマンスが向上し

ます。キャッシュ機能付きディスクコントローラーには予備電源装置を付け、停電時にもデータの整合性を確保できるようにする必要があります。

## ネットワークの性能

Session Recording Serverに接続するには100Mbpsのネットワークリンクが適しています。ギガビットイーサネット接続ではパフォーマンスが向上するかもしれませんが、100Mbpsのリンクの10倍のパフォーマンスが得られるわけではありません。

Session Recordingで使用するネットワークスイッチを、使用できるネットワーク帯域幅を求めて競合する可能性のあるサードパーティ製のアプリケーションと共有しないようにします。Session Recording Server専用のネットワークスイッチを用意することが理想的です。

## コンピューターの処理性能

Session Recording Serverをインストールするコンピューターについて、次の仕様を検討します。

- デュアルCPUまたはデュアルコアCPUをお勧めします。
- 2GBから4GBのRAMをお勧めします。

これらの仕様を超えるコンピューターを使用しても、パフォーマンスが大幅に向上することはありません。

## 複数のSession Recording Serverの展開

単一のSession Recording Serverではパフォーマンスのニーズを満たせない場合、複数のSession Recording Serverを異なるマシンにインストールできます。この種の展開では、各Session Recording Serverに専用のストレージ、ネットワークスイッチ、およびデータベースを設定します。負荷を分散するには、環境内の各Session Recording Agentで、異なるSession Recording Serverを参照するように設定します。

## データベースのスケラビリティ

Session Recordingデータベースでは、Microsoft SQL Server 2014、Microsoft SQL Server 2012、またはMicrosoft SQL Server 2008 R2が必要です。データベースにはセッション録画のメタデータのみが格納されるため、データベースに送信されるデータ量は非常に少なくなります。セッションの録画ファイル自体は別のディスクに書き込まれます。Session RecordingイベントAPIを使用してセッションに検索可能なイベントを挿入するのでなければ、セッション録画1件につきデータベースに必要な容量は通常1KBのみです。

Microsoft SQL Server 2014、Microsoft SQL Server 2012、およびMicrosoft SQL Server 2008 R2のExpress Editionでは、データベースサイズの上限は10GBです。1件のセッション録画あたり1KBのデータが書き込まれるとすれば、この制限があっても、4百万件のセッションをデータベースでカタログ化できます。Microsoft SQL Serverのほかのエディションではデータベースサイズの制限はなく、使用できるディスク容量によってのみ上限が決定されます。データベース内のセッション数が増加するにつれて、データベースのパフォーマンスと検索速度はごくわずかに低下します。

Session RecordingイベントAPIによるカスタマイズを行わない場合は、録画するセッションそれぞれについて、録画開始時に2件、ユーザーがセッションにログオンするときに1件、および録画終了時に1件の、合わせて4件のデータベーストランザクションが生成されます。Session RecordingイベントAPIを使用してセッションをカスタマイズする場合は、記録する検索可能なイベントそれぞれについて1件のトランザクションが生成されます。最も基本的な方式で展開したデータベースで、1秒当たり何百件というトランザクションを制御できるため、データベースの処理負荷が高くなる可能性はほとんどありません。影響が十分に小さいため、XenAppまたはXenDesktopのデータストアデータベースを含めたほかのデータベースと同じSQL Serverで、Session Recordingデータベースを実行できます。

Session Recordingのデータベースで何百万というセッション録画をカタログ化する必要がある場合は、SQL Serverのスケラビリティに関するMicrosoft社のガイドラインに従います。

# 展開に関する重要な注意事項

Jan 04, 2017

- Session Recordingコンポーネントを有効にして各コンポーネント間で通信できるようにするには、同じドメイン内か、推移的な信頼関係を持つ信頼されているドメイン間にインストールします。ワークグループまたは外部の信頼関係を持つドメイン間にはインストールできません。
- Session Recordingでは、単一の構成内で複数のSession Recording Serverをクラスター化することはサポートされていません。
- 映像を処理するアプリケーションであり、サイズの大きな録画を再生するときは多くのメモリが使用されるため、Session Recording Playerを公開アプリケーションとしてインストールすることはお勧めしません。
- デフォルトでは、Session RecordingはSSL/HTTPSを使用して通信するように構成されます。Session Recording Serverに証明書をインストールし、ルート証明機関 (CA) がSession Recordingコンポーネントで信頼されていることを確認します。
- Session RecordingデータベースをSQL Server 2014 Express Edition、SQL Server 2012 Express Edition、またはSQL Server 2008 R2 Express Editionが動作するスタンドアロンサーバーにインストールする場合は、そのサーバーでTCP/IPプロトコルを有効にしてSQL Server Browserサービスを実行する必要があります。これらの設定はデフォルトでは無効になっていますが、Session Recording Serverとデータベースの間で通信を行うために有効にする必要があります。これらの設定を有効にする方法については、Microsoftのドキュメントを参照してください。
- Session Recordingの展開を計画するときは、セッション共有の影響を考慮します。公開アプリケーションのセッションを共有すると、Session Recordingの公開アプリケーションの録画ポリシー規則と競合する可能性があります。Session Recordingでは、アクティブなポリシーとエンドユーザーが最初に開いた公開アプリケーションを照合します。エンドユーザーが最初のアプリケーションを開いた後で、同じセッション上で次のアプリケーションを開くと、最初のアプリケーションに対して有効なポリシーが、次のアプリケーションにも適用されます。たとえば、ポリシーがMicrosoft Outlookでの操作のみを録画する設定になっている場合、エンドユーザーがOutlookを開くと録画が始まります。しかし、Microsoft Outlookの実行中に公開アプリケーションのWordをエンドユーザーが開くと、Wordでの操作も録画されます。逆に、アクティブなポリシーがWordでの操作を録画する設定になっていない場合、エンドユーザーが (ポリシーに従って操作が録画されるべき) Outlookの前にWordを開くと、Outlookでの操作が録画されません。

# Session Recordingのインストール

Jan 04, 2017

この文書では、XenApp 7.6 Feature Pack 2およびLTSRのインストール手順と、Feature Pack 1のインストール手順についてセクションを分けて説明します。

## XenApp 7.6 Feature Pack 2およびLTSR

### インストール前のチェックリスト

インストールを始める前に、このリストに記載されている作業を行います。

|  | 作業                                                                                                         |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
|                                                                                   | インストールを開始する前に、前提条件をインストールします。「 <a href="#">システム要件</a> 」を参照してください。                                          |
|                                                                                   | Session Recordingの各コンポーネントをインストールするマシンを選択し、各コンピューターがインストールするコンポーネントのハードウェアおよびソフトウェアの要件を満たしていることを確認します。     |
|                                                                                   | LTSRのダウンロードページからSession RecordingのZIPファイルをダウンロードします。                                                       |
|                                                                                   | Session Recordingコンポーネント間の通信にSSLプロトコルを使用する場合は、正しい証明書を環境にインストールします。「 <a href="#">証明書のインストール</a> 」を参照してください。 |
|                                                                                   | Session Recordingコンポーネントに必要なHotfixをインストールします。Hotfixは <a href="#">Citrix Support</a> から入手できます。              |
|                                                                                   | Directorを構成して、Session Recordingポリシーを作成およびアクティブ化します。                                                        |

### 注：

- 公開アプリケーションのセッション共有は、同じデリバリーグループにあるとアクティブなポリシーと競合することがあるため、公開アプリケーションをSession Recordingポリシーをベースとした個別のデリバリーグループに分割することを推奨します。Session Recordingでは、アクティブなポリシーとエンドユーザーが最初に開いた公開アプリケーションを照合します。
- Machine Creation Services (MCS) またはProvisioning ServicesをXenAppと共に使用する計画がある場合は、一意なQMIDのサーバーを準備します。「[既知の問題](#)」の説明を参照してください。この手順の実行に失敗すると、記録データが損失することがあります。
- SQL ServerではTCP/IPを有効にする必要があります、SQL Server Browserサービスが実行中で、またWindows認証が必要です。
- HTTPSを使用する場合は、SSL/HTTPSのサーバー証明書を構成します。

### Session Recordingインストールファイル

Citrixのダウンロードページから次のインストールファイルをダウンロードする必要があります。

- Session Recording Administrationファイル
  - Broker\_PowerShellSnapIn\_x64.msi
  - SessionRecordingAdministrationx64.msiと置き換え
- Session Recording Agentファイル
  - SessionRecordingAgentx64.msiと置き換え
- Session Recording Playerファイル
  - SessionRecordingPlayer.msiと置き換え

## Session Recording Administrationコンポーネントのインストール

Session Recording Administrationコンポーネントは、Session Recordingデータベース、Session Recording Server、およびSession Recording Policy Consoleです。これらのコンポーネントのうち、サーバーにインストールするコンポーネントを選択できます。

Session Recording Administrationコンポーネントをインストールする前に、すべての前提条件をインストールしているか確認します。詳しくは、「[Session Recording Administrationコンポーネント](#)」を参照してください。

セキュリティ向上のため、データベースのインストール後にこれらの権限を削除できます。

1. **Broker\_PowerShellSnapIn\_x64.msi**を実行して、手順に従ってインストールを完了します。
2. Windowsコマンドプロンプトを管理者として起動し、次のコマンドを実行します。

```
msiexec /i SessionRecordingAdministrationx64.msi
```

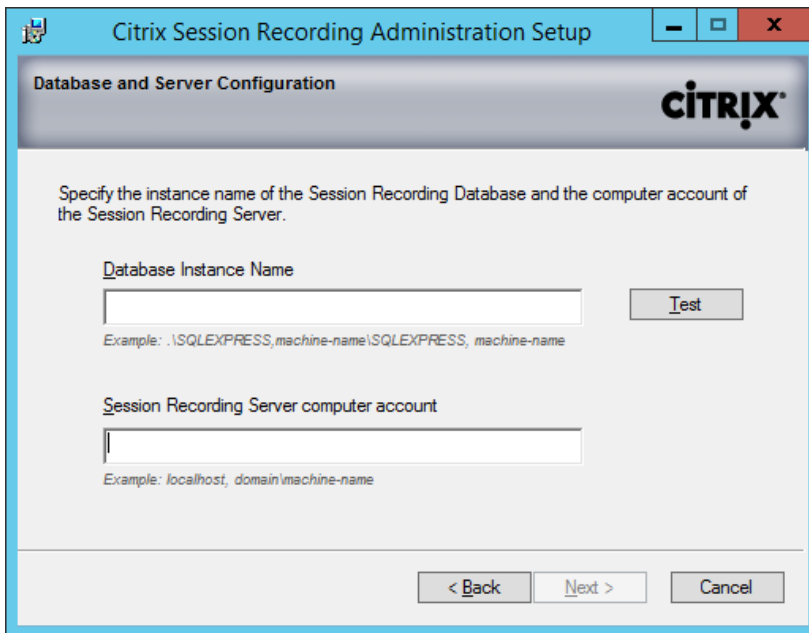
または、.msiファイルをダブルクリックします。

3. インストールUIで [次へ] を選択して、ライセンス契約に同意します。
4. Session Recording Administration Setup画面で、インストールするSession Recording Administrationコンポーネントを選択します。

## Session Recordingデータベースのインストール

Session Recordingデータベースをインストールする前に、すべての前提条件をインストールしているか確認します。詳しくは、「[Session Recording Administrationコンポーネント](#)」を参照してください。

1. [データベースの設定] ページで、次の作業を行います。
  - すべての管理コンポーネントを同じサーバーにインストールする場合は、[Session Recording Server Name] フィールドに「**localhost**」と入力します。
  - Session Recording ServerとSession Recordingデータベースを別のサーバーにインストールする場合は、Session Recording Serverをホストするコンピューターの名前を「ドメイン\マシン名」の形式で指定します。Session Recording Server名はデータベースにアクセスするためのユーザーアカウントです。



データベースインスタンスが、インスタンスのセットアップ時に構成した名前付きインスタンスでない場合、SQL Serverのマシン名のみを使用できます。名前付きインスタンスがある場合は、データベースインスタンス名としてmachine-name\instance-nameを使用します。使用中のサーバーインスタンス名を確認するには、SQL Serverで**select @@servername**を実行します。戻り値が正確なデータベースインスタンス名です。

【テスト】をクリックして、SQL Serverへの接続をテストします。現在のユーザーがSQL Serverのパブリックな役割権限を持っていることを確認します。権限がなければ、権限の制約によりテストは失敗します。【次へ】をクリックしてインストールを続行します。

手順に従ってインストールを完了します。インストール中、現在のユーザーがデータベース管理者ではない場合、**sysadmin**サーバー役割権限を持つデータベース管理者の資格情報を求めるダイアログボックスが表示されます。資格情報を正確に入力して、【OK】をクリックし、インストールを続行します。インストールにより新しいSession Recordingデータベースが作成され、Session Recording Serverのマシニアカウントが**db-owner**として追加されます。

### Session Recording Serverのインストール

Session Recording Serverをインストールする前に、すべての前提条件をインストールしているか確認します。詳しくは、「[Session Recording Administrationコンポーネント](#)」を参照してください。

1. 【データベースインスタンス名】にSQL Serverの名前を入力します。名前付きインスタンスを使用している場合は、「machine-name\instance-name」と入力します。使用していない場合は、「machine-name」だけを入力します。
2. 【テスト】をクリックして、SQL Serverへの接続をテストします。現在のユーザーがSQL Serverのパブリックな役割権限を持っていることを確認します。権限がなければ、権限の制約によりテストは失敗します。【次へ】をクリックし、ウィザードの指示に従ってインストールを完了します。
3. インストールウィザードの最後に、Citrixのカスタマーエクスペリエンス向上プログラムに参加するかどうかを選択します。このプログラムに参加すると、統計情報や使用状況が匿名でCitrixに送信されます。詳しくは、「[Citrixカスタマーエクスペリエンス向上プログラム \(CEIP\) について](#)」を参照してください。

### Session Recording Agentのインストール

Session Recording Agentはセッションを録画するサーバーOS VDAマシンにインストールする必要があります。

1. サーバーマネージャーを使って.NET Framework 3.5とXenApp 7.6 Server OS VDAでHTTPをサポートするMicrosoft

Message Queuing (MSMQ) をインストールします。

2. Windowsコマンドプロンプトを管理者として起動し、次のコマンドを実行します。

```
msiexec /i SessionRecordingAgentx64.msi
```

または、.msiファイルをダブルクリックします。

3. インストールUIで [次へ] を選択して、ライセンス契約に同意します。

4. [Session Recording Agent構成] ページで、Session Recording Serverをインストールしたコンピューターの名前と、Session Recording Serverとの接続のプロトコルとポート情報を入力します。

Session Recordingのデフォルトのインストールでは、通信をセキュリティで保護するためHTTPS/SSLを使用します。SSLが構成されていない場合は、HTTPを使用します。これを実行するには、Session Recording Brokerサイトに移動してIIS管理コンソールでSSLを選択解除します。SSL設定を開き、[SSLを必要とする] チェックボックスをオフにします。

5. Feature Pack 2でロールオーバー機能を使用する場合は、Update ICATS760WX64010 (<http://support.citrix.com/article/CTX142037>) をインストールします。LTSRの場合では、ロールオーバー機能はLTSRバージョンのVDA (7.6.300) に含まれています。このため、VDAがLTSRレベルである場合、特に操作をすることなくこの機能を使用できます。

6.手順に従ってインストールを完了します。

### Session Recording Playerのインストール

Session Recording Serverまたはセッションの録画を見るユーザーのドメイン内の1つ以上のワークステーションにSession Recording Playerをインストールします。

- **SessionRecordingPlayer.msi**を実行して、手順に従ってインストールを完了します。

### Session Recordingのアップグレード

Session RecordingをXenApp 7.6 Feature Pack 1または2からアップグレードするには、対応するSession Recording機能がインストールされているマシンでインストーラーを実行します。

手順に従って、アップグレードのインストールを完了します。アップグレードの処理中、情報の入力はありません。

## XenApp 7.6 Feature Pack 1

### インストール前のチェックリスト

インストールを始める前に、このリストに記載されている作業を行います。

|  | 作業                                                                                                         |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
|                                                                                   | インストールを開始する前に、前提条件をインストールします。 <a href="#">「システム要件」</a> を参照してください。                                          |
|                                                                                   | Session Recordingの各コンポーネントをインストールするマシンを選択し、各コンピューターがインストールするコンポーネントのハードウェアおよびソフトウェアの要件を満たしていることを確認します。     |
|                                                                                   | <b>[XenApp] &gt; [コンポーネント]</b> にあるCitrixダウンロードページからSession RecordingのZIPファイルをダウンロードします。                    |
|                                                                                   | Session Recordingコンポーネント間の通信にSSLプロトコルを使用する場合は、正しい証明書を環境にインストールします。 <a href="#">「証明書のインストール」</a> を参照してください。 |
|                                                                                   | Session Recordingコンポーネントに必要なHotfixをインストールします。 Hotfixは <a href="#">Citrix Support</a> から入手できます。             |
|                                                                                   | Directorを構成して、Session Recordingポリシーを作成およびアクティブ化します。                                                        |

### 注：

- 公開アプリケーションのセッション共有は、同じデリバリーグループにあるとアクティブなポリシーと競合することがあるため、公開アプリケーションをSession Recordingポリシーをベースとした個別のデリバリーグループに分割することを推奨します。Session Recordingでは、アクティブなポリシーとエンドユーザーが最初に開いた公開アプリケーションを照合します。
- Machine Creation Services (MCS) またはProvisioning ServicesをXenAppと共に使用する計画がある場合は、一意なQMIDのサーバーを準備します。「既知の問題」の説明を参照してください。この手順の実行に失敗すると、記録データが損失することがあります。
- SQL ServerではTCP/IPを有効にする必要があり、SQL Server Browserサービスが実行中で、またWindows認証が必要です。
- HTTPSを使用する場合は、SSL/HTTPSのサーバー証明書を構成します。

### Session Recordingインストールファイル

Citrixのダウンロードページから次のインストールファイルをダウンロードする必要があります。

- Session Recording Administrationファイル
  - Broker\_PowerShellSnapIn\_x64.msi
  - SessionRecordingAdministrationx64.msiと置き換え



- Session Recording Agentファイル
  - SessionRecordingAgentx64.msiと置き換え
- Session Recording Playerファイル
  - SessionRecordingPlayer.msiと置き換え

## Session Recording Administrationコンポーネントのインストール

Session Recording Administrationコンポーネントは、Session Recordingデータベース、Session Recording Server、およびSession Recording Policy Consoleです。これらのコンポーネントのうち、サーバーにインストールするコンポーネントを選択できます。

Session Recording Administrationコンポーネントをインストールする前に、すべての前提条件をインストールしているか確認します。詳しくは、「[Session Recording Administrationコンポーネント](#)」を参照してください。

ローカルのSQL ServerにSession Recordingデータベースをインストールする場合、SQL Serverの役割権限ysadminでログインするときはNT AUTHORITY\SYSTEMが必要です。

リモートのSQL ServerにSession Recordingデータベースをインストールする場合、SQL Serverの役割権限ysadminでログインするときはインストーラーを実行しているマシンアカウントが必要です。

セキュリティ向上のため、データベースのインストール後にこれらの権限を削除できます。

1. Broker\_PowerShellSnapIn\_x64.msiを実行して、手順に従ってインストールを完了します。
2. Windowsコマンドプロンプトを管理者として起動し、次のコマンドを実行します。

```
msiexec /i SessionRecordingAdministrationx64.msi
```

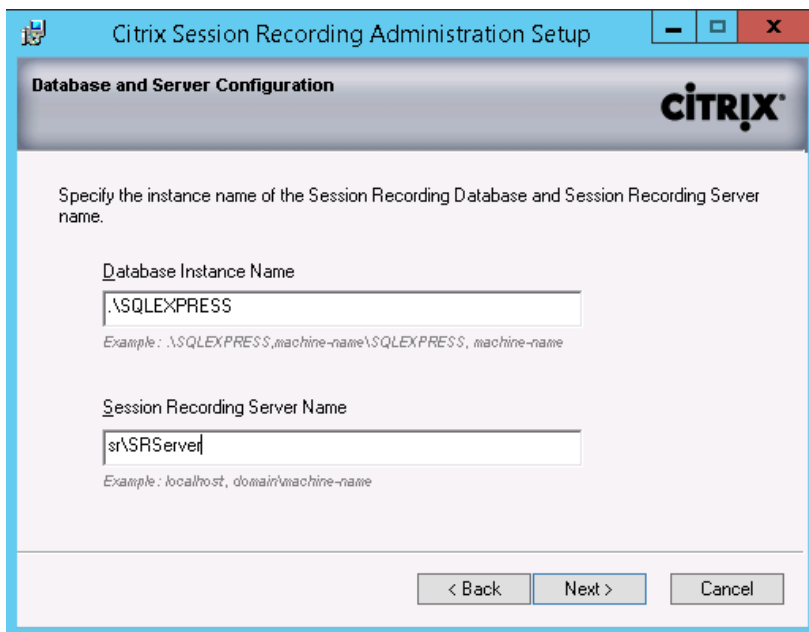
または、.msiファイルをダブルクリックします。

3. インストールUIで[次へ]を選択して、ライセンス契約に同意します。
4. Session Recording Administration Setup画面で、インストールするSession Recording Administrationコンポーネントを選択します。

## Session Recordingデータベースのインストール

Session Recordingデータベースをインストールする前に、すべての前提条件をインストールしているか確認します。詳しくは、「[Session Recording Administrationコンポーネント](#)」を参照してください。

1. [データベースの設定] ページで、次の作業を行います。
  - すべての管理コンポーネントを同じサーバーにインストールする場合は、[Session Recording Server Name] フィールドに「**localhost**」と入力します。
  - Session Recording ServerとSession Recordingデータベースを別のサーバーにインストールする場合は、Session Recording Serverをホストするコンピューターの名前を「ドメイン\マシン名」の形式で指定します。Session Recording Server名はデータベースにアクセスするためのユーザーアカウントです。



データベースインスタンスが、インスタンスのセットアップ時に構成した名前付きインスタンスでない場合、SQL Serverのマシン名のみを使用できます。名前付きインスタンスがある場合は、データベースインスタンス名として *machine-name\instance-name* を使用します。使用中のサーバーインスタンス名を確認するには、SQL Serverで `select @@servername` を実行します。戻り値が正確なデータベースインスタンス名です。

2. 手順に従ってインストールを完了します。インストールにより新しいSession Recordingデータベースが作成され、Session Recording Serverのマシンアカウントが **db-owner** として追加されます。

## Session Recording Serverのインストール

Session Recording Serverをインストールする前に、すべての前提条件をインストールしているか確認します。詳しくは、「[Session Recording Administrationコンポーネント](#)」を参照してください。

- [データベースインスタンス名] にSQL Serverの名前を入力します。名前付きインスタンスを使用している場合は、「*machine-name\instance-name*」と入力します。使用していない場合は、「*machine-name*」だけを入力します。

## Session Recording Agentのインストール

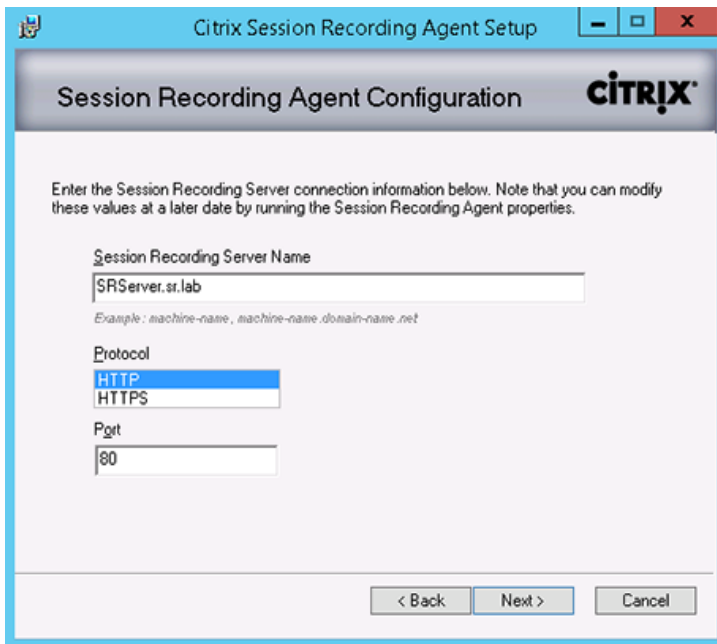
Session Recording Agentはセッションを録画するサーバーOS VDAマシンにインストールする必要があります。

1. サーバーマネージャーを使って.NET Framework 3.5とXenApp 7.6 Server OS VDAでHTTPをサポートするMicrosoft Message Queuing (MSMQ) をインストールします。
2. Windowsコマンドプロンプトを管理者として起動し、次のコマンドを実行します。

```
msiexec /i SessionRecordingAgentx64.msi
```

または、.msiファイルをダブルクリックします。

3. インストールUIで [次へ] を選択して、ライセンス契約に同意します。
4. [Session Recording Agent構成] ページで、Session Recording Serverをインストールしたコンピューターの名前と、Session Recording Serverとの接続のプロトコルとポート情報を入力します。



Session Recordingのデフォルトのインストールでは、通信をセキュリティで保護するためHTTPS/SSLを使用します。SSLが構成されていない場合は、HTTPを使用します。これを実行するには、Session Recording Brokerサイトに移動してIIS管理コンソールでSSLを選択解除します。SSL設定を開き、[SSLを必要とする] チェックボックスをオフにします。

5. ロールオーバー機能を使用する場合は、Hotfix <http://support.citrix.com/article/CTX142037> をインストールしていることを確認します。
6. 手順に従ってインストールを完了します。

### Session Recording Playerのインストール

Session Recording Serverまたはセッションの録画を見るユーザーのドメイン内の1つ以上のワークステーションにSession Recording Playerをインストールします。

- SessionRecordingPlayer.msiを実行して、手順に従ってインストールを完了します。

### Session Recordingのアンインストール

サーバーやワークステーションからSession Recordingコンポーネントを削除するには、Windowsのコントロールパネルのプログラムのアンインストールまたは削除機能を使用します。Session Recordingデータベースを削除するには、インストールと同じSQL Serverの役割権限sysadminが必要です。

### Directorを構成してSession Recording Serverを使用する

Directorコンソールを使用して、Session Recordingポリシーを作成およびアクティブ化できます。

1. HTTPS接続の場合は、Directorサーバーの[信頼されたルート証明機関]でSession Recording Serverを信頼します。
2. Session Recording Serverを使用するようにDirectorサーバーを構成するには、次のコマンドを実行します。  
`C:\inetpub\wwwroot\Director\tools\DirectorConfig.exe /configsessionrecording`
3. Directorサーバーで、Session Recording ServerのIP/FQDN、ポート番号、およびSession Recording AgentからSession Recording Brokerへの接続の種類 (httpまたはhttps) を入力します。

## インストールの自動化

Jan 04, 2017

Session Recording Agentを複数のサーバーにインストールするには、サイレントインストールを行うスクリプトを作成します。

次のコマンドラインでは、Session Recording Agentをインストールし、インストール情報を取得するためにログファイルを作成します。

```
msiexec /i SessionRecordingAgentx64.msi sessionrecordingservername=<yourservname> sessionrecordingbrokerprotoco=<yourbrokerprotocol> sessionrecordingbrokerport=<yourbrokerport> /!v <you
```

各項目の意味は次のとおりです。

<yourservname>は、Session Recording ServerをホストするコンピューターのNetBIOS名またはFQDNです。指定しない場合のデフォルト値はlocalhostです。

<yourbrokerprotocol>は、HTTPまたはHTTPSで、Session Recording Brokerとの通信にSession Recording Agentで使用されるプロトコルです。指定しない場合のデフォルト値はHTTPSです。

<yourbrokerport>は、Session Recording Brokerとの通信にSession Recording Agentで使用されるポートを表す整数です。指定しない場合のデフォルト値は0で、選択したプロトコルのデフォルトのポート番号、つまりHTTPでは80、HTTPSでは443がSession Recording Agentで使用されます。

/!vスイッチにより詳細モードでログが記録されます。

<urinstallationlog>は、セットアップログを作成する場所です。

/qスイッチによりサイレントモードでインストールされます。

# セッションの再生と録画のためのSession Recordingの構成

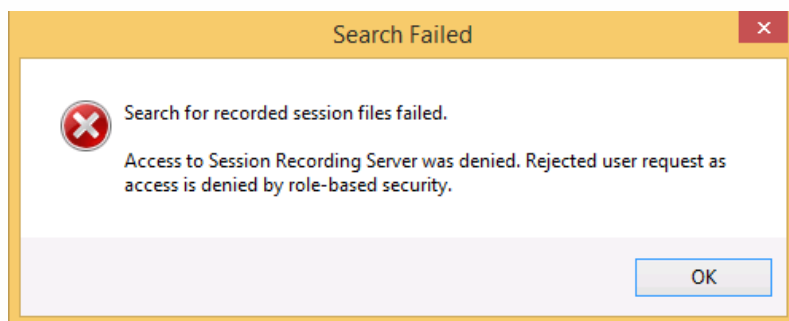
Jan 04, 2017

Session Recordingコンポーネントをインストールしたら、次の作業を行いSession Recordingを設定して、XenAppセッションを録画して表示できるようにします。

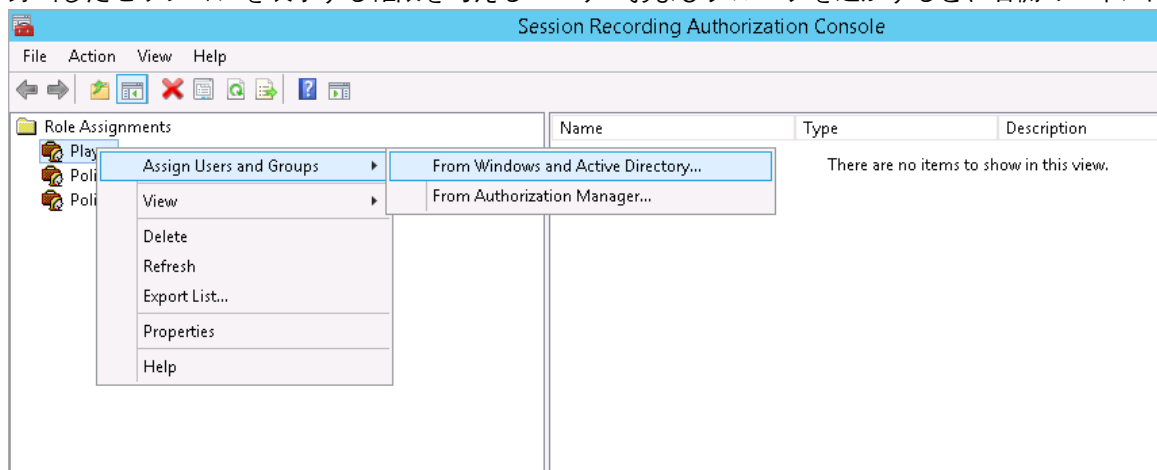
- ユーザーに録画を再生する権限を与える。
- ユーザーに録画ポリシーを管理する権限を与える。
- セッションを録画する録画ポリシーをアクティブにする。
- Session Recording Playerを構成してSession Recording Serverと接続する。

ユーザーに録画したセッションを再生する権限を与える。

Session Recordingのデフォルト設定では、どのユーザーにも録画したセッションを再生する権限はありません。管理者も含め、各ユーザーに権限を割り当てる必要があります。録画したセッションを再生する権限がないユーザーが録画したセッションを再生しようとすると次のエラーメッセージが表示されます。



1. Session Recording Serverをホストするコンピューターに管理者としてログオンします。
2. Session Recording承認コンソールを起動します。
3. Session Recording承認コンソールで [Player] を選択します。
4. 録画したセッションを表示する権限を与えるユーザーおよびグループを追加すると、右側のペインに表示されます。



ユーザーに録画ポリシーを管理する権限を与える。

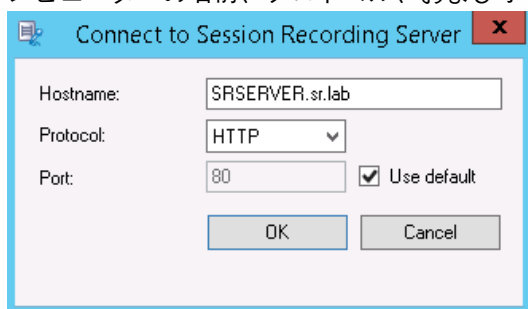
Session Recordingをインストールすると、ドメイン管理者から、録画ポリシーを制御する権限がデフォルトで与えられます。承認設定は変更できます。

1. Session Recording Serverをホストするマシンに管理者としてログオンします。
2. Session Recording承認コンソールを起動して、PolicyAdministratorsを選択します。
3. 録画ポリシーを管理できるユーザーとグループを追加します。

セッションを録画する録画ポリシーをアクティブに設定。

アクティブな録画ポリシーにより、Session Recording AgentがインストールされておりSession Recording Serverに接続する、すべてのサーバーOS VDAでのセッションの録画処理が決定されます。Session Recordingのデフォルト設定では、アクティブな録画ポリシーは「録画しない」ポリシーです。アクティブな録画ポリシーを変更するまで、セッションは録画できません。

1. 承認済みのポリシー管理者として、Session Recording Policy Consoleがインストールされているサーバーにログオンします。
2. Session Recording Policy Consoleを起動します。
3. [Connect to Session Recording Server] ポップアップウィンドウが開いたら、Session Recording Serverをホストするコンピューターの名前、プロトコル、およびポートが正しいことを確認します。



4. Session Recording Policy Consoleで、[録画ポリシー]を展開します。Session Recordingの標準の録画ポリシーが表示されます。チェックマークはアクティブなポリシーを示します。
  - 録画しない。これがデフォルトのポリシーです。ほかのポリシーを指定しなければ、セッションは録画されません。
  - すべてのユーザーを通知して録画する。このポリシーを選択すると、すべてのセッションが録画されます。録画されていることを通知するポップアップウィンドウがエンドユーザーに表示されます。
  - すべてのユーザーを通知しないで録画する。このポリシーを選択すると、すべてのセッションが録画されます。録画されていることを通知するポップアップウィンドウが開くことはありません。
5. アクティブにするポリシーを選択します。
6. メニューバーで [操作] の [ポリシーのアクティブ化] を選択します。

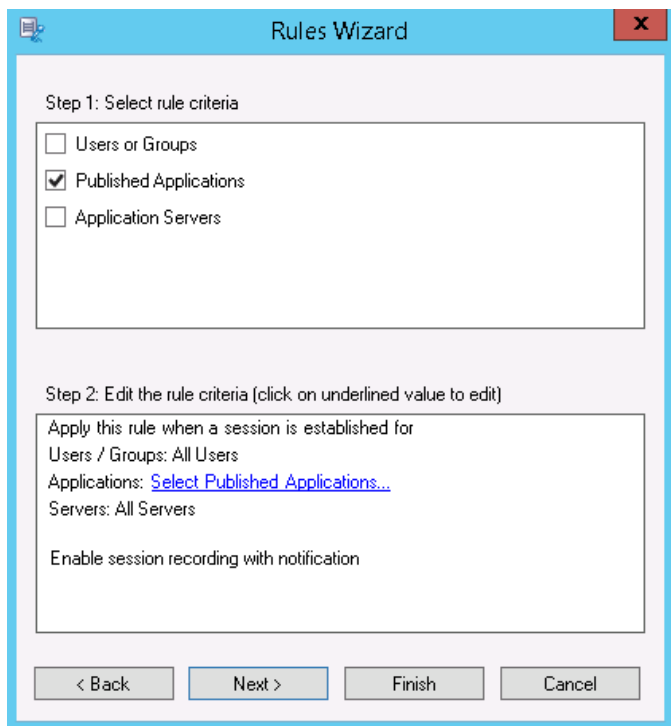
注：Session Recordingでは独自の録画ポリシーを作成できます。録画ポリシーを作成すると、Session Recording Policy Consoleの [録画ポリシー] フォルダーに表示されます。

#### カスタムポリシーの構成

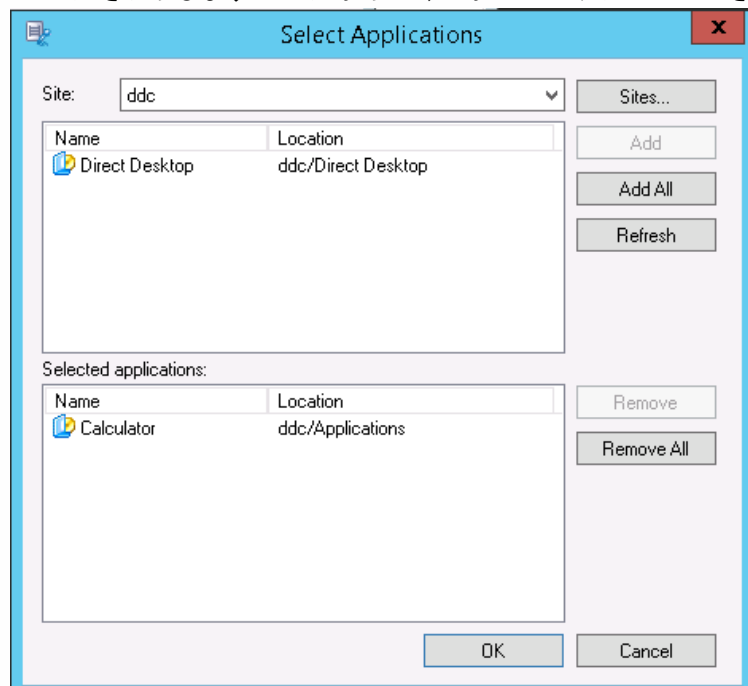
一般的な録画ポリシーでは要件に合わないことがあります。ユーザー、VDAサーバー、およびアプリケーションをベースとしてポリシーを構成できます。

**重要：**ポリシーには多くの規則を含めることができますが、一度にアクティブにできるポリシーは1つだけです。

1. 承認済みのポリシー管理者として、Session Recording Policy Consoleがインストールされているサーバーにログオンします。
2. Session Recording Policy Consoleを起動して [レコーディングポリシー] の [新しいポリシーの追加] を選択します。
3. [新しいポリシー] を右クリックして [新しい規則の追加] を選択します。
4. 規則ウィザードで、[通知してセッションを録画する] を選択し [次へ] をクリックします。
5. [公開アプリケーション] チェックボックスをオンにして、[公開アプリケーションを選択] ハイパーリンクをクリックします。



6. [アプリケーションの選択] 画面で、[サイト] および [追加] をクリックします。
7. XenApp 7.6 FP1 Delivery Controllerのサーバー名を入力します。
8. [接続] をクリックしてサイトを列挙します。
9. [閉じる] をクリックし、公開アプリケーションの一覧を表示します。一覧からいくつかのアプリケーションを追加



し、[OK] および [完了] をクリックします。

10. ポリシーを右クリックして [アクティブ化] を選択します。必要な場合はポリシーの名前を変更できます。

### Session Recording Playerの構成

Session Recording Playerでセッションを再生するには、録画されたセッションを格納するSession Recording Playerとの接続を設定する必要があります。Session Recording Playerごとに複数のSession Recording Serverとの接続を設定できますが、同時に複数のSession Recording Serverに接続することはできません。Session Recording Playerに複数のSession Recording

Serverとの接続が構成される場合は、チェックボックスを使用して接続先のSession Recording Serverを変更できます。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. Session Recording Playerを起動します。
3. Session Recording Playerのメニューバーで [ツール] > [オプション] の順に選択します。
4. [接続] タブで [追加] をクリックします。
5. [ホスト名] フィールドに、Session Recording Serverをホストするコンピューターの名前かインターネットプロトコル (IP) アドレスを入力し、プロトコルを選択します。デフォルトでは、セキュリティで保護された通信のためHTTPS/SSLを使用するようにSession Recordingが構成されます。SSLが構成されていない場合は、HTTPを選択します。
6. Session Recording Playerが複数のSession Recording Serverと接続できるように構成する場合は、Session Recording Serverごとに手順4および5を繰り返します。
7. 接続するSession Recording Serverのチェックボックスがオンになっていることを確認します。



# ユーザーへのアクセス権の付与

Jan 04, 2017

注：セキュリティ上の理由から、セッションの録画の表示など、特定の機能を実行するために必要な権限のみをユーザーに付与します。

Session Recording ServerをホストするコンピューターでSession Recording承認コンソールを使用して役割を割り当てることにより、Session Recordingユーザーに権限を付与します。Session Recordingユーザーには3つの役割があります。

- **Player**。録画したXenAppセッションを表示できます。デフォルトでこの役割のメンバーになるユーザーはありません。
- **PolicyQuery**。Session Recording Agentをホストするサーバーで録画ポリシーの評価を要求できます。デフォルトでは、Authenticated Usersがこの役割のメンバーです。
- **PolicyAdministrator**。録画ポリシーの表示、作成、編集、削除、および有効化を実行できます。デフォルトでは、Session Recording Serverをホストするコンピューターの管理者がこの役割のメンバーです。

Session Recordingでは、Active Directoryで定義されるユーザーおよびグループがサポートされます。

ユーザーに役割を割り当てるには

1. Session Recording Serverをホストするコンピューターに、管理者またはPolicyAdministratorのメンバーとしてログオンします。
2. Session Recording承認コンソールを起動します。
3. ユーザーを割り当てる役割を選択します。
4. [操作]メニューの[Windowsユーザーとグループの割り当て]を選択します。
5. ユーザーとグループを追加します。

管理コンソールで加えた変更は、1分間隔の更新時に有効になります。

# 録画ポリシーの作成とアクティブ化

Jan 04, 2017

Session Recording Policy Consoleを使用して、録画するセッションを決定するポリシーを作成しアクティブにします。

Session Recordingをインストールすると使用できるようになるシステムポリシーをアクティブにすることも、独自のカスタムポリシーを作成してアクティブにすることもできます。Session Recordingのシステムポリシーでは、すべてのユーザー、公開アプリケーション、およびサーバーに、単一の規則を適用します。カスタムポリシーでは、録画対象のユーザー、公開アプリケーション、およびサーバーを指定します。

アクティブなポリシーによって録画するセッションが決定されます。一度にアクティブにできるポリシーは1つだけです。

## システムポリシーの使用

Session Recordingには、次の3つのシステムポリシーがあります。

- **録画しない。** このポリシーを選択すると、セッションは録画されません。これがデフォルトのポリシーです。ほかのポリシーを指定しなければ、セッションは録画されません。
- **すべてのユーザーを通知して録画する。** このポリシーを選択すると、すべてのセッションが録画されます。録画されていることを通知するポップアップウィンドウがエンドユーザーに表示されます。
- **すべてのユーザーを通知しないで録画する。** このポリシーを選択すると、すべてのセッションが録画されます。録画されていることを通知するポップアップウィンドウが開くことはありません。

システムポリシーは変更したり削除したりできません。

## ポリシーをアクティブにするには

1. Session Recording Policy Consoleがインストールされているサーバーにログオンします。
2. Session Recording Policy Consoleを起動します。
3. [Connect to Session Recording Server] ポップアップウィンドウが開いたら、Session Recording Serverの名前、プロトコル、およびポートが正しいことを確認します。[OK] をクリックします。
4. Session Recording Policy Consoleで、[録画ポリシー] を展開します。
5. アクティブにするポリシーを選択します。
6. メニューバーで [操作] の [ポリシーのアクティブ化] を選択します。

## カスタム録画ポリシーの作成

独自のポリシーを作成するときは、規則を作成して、録画対象のユーザーおよびグループ、公開アプリケーション、およびサーバーを指定します。Session Recording Policy Consoleにはウィザードが用意されており、このウィザードに従って規則を作成します。公開されているアプリケーションやサーバーの一覧を取得するには、サイト管理者の読み取り権限が必要です。この権限はこのサイトのDelivery Controllerで設定します。

作成する規則ごとに録画操作および規則条件を指定します。録画操作は規則条件を満たすセッションに適用されます。

規則ごとに録画操作を1つ選択します。

- **録画しない。** (規則ウィザードで [セッションを録画しない] をクリックします) この録画操作では、規則条件を満たすセッションを録画しないことを指定します。
- **通知して録画する。** (規則ウィザードで [通知してセッションを録画する] をクリックします) この録画操作では、規則条件を満たすセッションを録画することを指定します。録画されていることを通知するポップアップウィンドウがエンドユーザーに表示されます。

- 通知しないで録画する。(規則ウィザードで [通知しないでセッションを録画する] をクリックします) この録画操作では、規則条件を満たすセッションを録画することを指定します。エンドユーザーは録画されていることに気が付きません。

規則ごとに次のいずれかを少なくとも1つ選択して、規則条件を作成します。

- ユーザーまたはグループ。規則の録画操作を適用するユーザーまたはグループの一覧を作成します。
- 公開アプリケーション。規則の録画操作を適用する公開アプリケーションの一覧を作成します。規則ウィザードで、アプリケーションを使用できる1つまたは複数のXenAppサイトを選択します。
- アプリケーションサーバー。規則の録画操作を適用するサーバーOSマシンの一覧を作成します。規則ウィザードで、サーバーが属する1つまたは複数のXenAppサイトを選択します。

録画ポリシーに複数の規則を作成する場合は、複数の規則条件に一致するセッションがある可能性があります。そのような場合は、優先順位が最も高い規則がセッションに適用されます。

規則により実行される録画操作によってその優先順位が決まります。

- 録画しない規則の優先順位が最も高くなります。
- 通知して録画する規則の優先順位が次に高くなります。
- 通知しないで録画する規則の優先順位が最も低くなります。

録画ポリシーの規則条件のいずれにも当てはまらないセッションがある可能性があります。そのようなセッションについては、フォールバック規則の録画操作が適用されます。フォールバック規則の録画操作は常に「録画しない」です。フォールバック規則は変更したり削除したりできません。

### Active Directoryグループの使用

Active Directoryグループを使用して、Session Recordingのポリシーを作成できます。個々のユーザーではなくActive Directoryグループを使用すると、規則とポリシーを簡単に作成したり管理したりできます。たとえば、財務部門のユーザーがFinanceという名前のActive Directoryグループに含まれている場合は、規則を作成するときに規則ウィザードでFinanceグループを選択することで、このグループのすべてのメンバーに適用される規則を作成できます。

### ユーザーのホワイトリスト化

組織内の一部のユーザーのセッションを確実に録画対象から除外する、Session Recordingポリシーを作成できます。これはユーザーのホワイトリスト化と呼ばれます。個人情報を取り扱う社員や特定の階層の従業員など、セッションを録画するべきではないユーザーをホワイトリストに登録すると便利です。

すべての上級管理職がExecutiveという名前のActive Directoryグループのメンバーである場合、Executiveグループのセッション録画を無効にする規則を作成して、それらのユーザーのセッションが決して録画されないように設定できます。この規則を含むポリシーがアクティブな間は、Executiveグループのメンバーのセッションは録画されません。組織内のほかのメンバーのセッションは、アクティブなポリシーのほかの規則に基づいて録画されます。

### 新しいポリシーの作成

注：規則ウィザードで、下線付きの値が表示されていないにもかかわらず「下線が引かれている値をクリックして編集できません」と表示される場合があります。このウィザードでは、適用される場合のみ下線付きの値が表示されます。下線付きの値が表示されない場合は、この手順を無視して次に進んでください。

1. Session Recording Policy Consoleがインストールされているサーバーにログオンします。
2. Session Recording Policy Consoleを起動します。
3. [Connect to Session Recording Server] ポップアップウィンドウが開いたら、Session Recording Serverの名前、プロトコル、およびポートが正しいことを確認します。[OK] をクリックします。
4. Session Recording Policy Consoleで、[録画ポリシー] を選択します。

5. メニューバーで [操作] の [新しいポリシーの追加] を選択します。「新しいポリシー」という名前のポリシーが左ペインに表示されます。
6. 新しいポリシーを選択し、メニューバーで [操作] の [名前の変更] を選択します。
7. 作成したポリシーの名前を入力し、Enterキーを押すか新しい名前の外側の任意の場所をクリックします。
8. ポリシーを選択した状態で、メニューバーで [操作] の [新しい規則の追加] を選択し、規則ウィザードを起動します。
9. 画面の指示に従ってこのポリシーの規則を作成します。

## ポリシーの変更

1. Session Recording Policy Consoleがインストールされているサーバーにログオンします。
2. Session Recording Policy Consoleを起動します。
3. [Connect to Session Recording Server] ポップアップウィンドウが開いたら、Session Recording Serverの名前、プロトコル、およびポートが正しいことを確認します。 [OK] をクリックします。
4. Session Recording Policy Consoleで、 [録画ポリシー] を展開します。
5. 変更するポリシーを選択します。ポリシーの規則が右ペインに表示されます。
6. 新しい規則を追加したり、規則を変更または削除したりします。
  - メニューバーで [操作] の [新しい規則の追加] を選択します。ポリシーがアクティブな場合はポップアップウィンドウが開き、操作の確認を促すメッセージが表示されます。規則ウィザードを使用して新しい規則を作成します。
  - 変更する規則を選択し、右クリックして [プロパティ] を選択します。規則ウィザードを使用して規則を変更します。
  - 削除する規則を選択し、右クリックして [規則の削除] を選択します。

## ポリシーの削除

注：システムポリシーまたはアクティブなポリシーは削除できません。

1. Session Recording Policy Consoleがインストールされているサーバーにログオンします。
2. Session Recording Policy Consoleを起動します。
3. [Connect to Session Recording Server] ポップアップウィンドウが開いたら、Session Recording Serverの名前、プロトコル、およびポートが正しいことを確認します。 [OK] をクリックします。
4. Session Recording Policy Consoleで、 [録画ポリシー] を展開します。
5. 左ペインで削除するポリシーを選択します。ポリシーがアクティブな場合は、ほかのポリシーをアクティブにする必要があります。
6. メニューバーで [操作] の [ポリシーの削除] を選択します。
7. [はい] をクリックして操作を確定します。

## ロールオーバーの動作

ポリシーをアクティブにするとき、それまでアクティブだったポリシーはユーザーセッションが終了するまで効力を保ちます。ただし、ファイルがロールオーバーするときに新しいポリシーが有効になる場合があります。ファイルサイズが上限にするとロールオーバーが実行されます。録画ファイルのサイズの上限については、「[録画ファイルのサイズの指定](#)」を参照してください。

次の表で、セッションの録画中に新しいポリシーを適用してロールオーバーが起きたときに生じる現象について説明します。

| 以前のポリシー    | 新しいポリシー | ロールオーバーの後                                         |
|------------|---------|---------------------------------------------------|
| 録画しない      | ほかのポリシー | 変更なし。エンドユーザーが新しいセッションにログオンするときのみに新しいポリシーが有効になります。 |
| 通知しないで録画する | 録画しない   | 録画を停止します。                                         |

|          |            |                                            |
|----------|------------|--------------------------------------------|
| 以前のポリシー  | 通知して録画する   | 録画を続行し、通知メッセージを表示します。                      |
|          | 新しいポリシー    | 録画を停止し、メッセージを表示します。                        |
| 通知して録画する | 録画しない      | 録画を停止します。                                  |
|          | 通知しないで録画する | 録画を続行します。エンドユーザーが次にログオンするときはメッセージが表示されません。 |

# 録画の無効化または有効化

Jan 04, 2017

Session Recording Agentは、セッションを録画する各サーバーOSマシンにインストールします。インストール先のサーバーで録画を有効にするかどうかの設定を、Session Recording Agentで行います。録画を有効にすると、Session Recordingによりアクティブな録画ポリシーが評価されます。このポリシーにより録画対象のセッションが決定されます。

Session Recording Agentをインストールすると、デフォルトで録画が有効になります。録画しないサーバーではSession Recordingを無効にすることをお勧めします。録画をしていないときでも、サーバーのパフォーマンスが多少影響を受けるためです。

サーバーで録画を無効または有効にするには

1. Session Recording Agentがインストールされているサーバーにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Agent] を選択します。
3. [セッションの録画] で [このXenAppサーバーでセッションを録画する] チェックボックスをオンまたはオフにし、このサーバーでセッションを録画できるようにするかどうかを指定します。
4. 確認メッセージが表示されるので、Session Recording Agentサービスを再起動して変更を受け入れます。

注：Session Recordingをインストールしたときにアクティブになっているポリシーは「録画しない」ポリシーです。どのサーバーのセッションも録画されません。録画を開始するには、Session Recording Policy Consoleを使用して別のポリシーをアクティブにします。

# Session Recording Serverとの接続の設定

Jan 04, 2017

Session Recording AgentとSession Recording Serverの間の接続は、通常、Session Recording Agentをインストールするときに設定します。Session Recording Agentをインストールした後でこの接続を設定するには、[Session Recording Agentのプロパティ] を使用します。

1. Session Recording Agentがインストールされているサーバーにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Agentのプロパティ] を選択します。
3. [接続] タブをクリックします。
4. [Session Recording Server] フィールドに、サーバーの名前かインターネットプロトコル (IP) アドレスを入力します。
5. [Session Recordingストレージマネージャーメッセージのキュー] で、Session Recordingストレージマネージャーが通信に使用するプロトコルを選択し、必要であればデフォルトのポート番号を変更します。
6. [メッセージの有効期間] ボックスで、通信エラーが発生したときにキューに各メッセージを保持する秒数として、デフォルトの7200秒 (2時間) を受け入れるか、新しい値を入力します。この期間が経過すると、メッセージは削除され、ファイルを再生できるのはデータが失われた時点までになります。
7. [Session Recording Broker] で、Session Recording Brokerが通信に使用するプロトコルを選択し、必要であればデフォルトのポート番号を変更します。
8. 確認メッセージが表示されるので、Session Recording Agentサービスを再起動して変更を受け入れます。

# 通知メッセージの作成

Jan 04, 2017

アクティブな録画ポリシーにより、セッションを録画するときにユーザーに通知する設定になっている場合、ユーザーが資格情報を入力した後にポップアップウィンドウが開き、通知メッセージが表示されます。デフォルトの通知メッセージは「現在開始しているプログラムでの操作が録画されています。この条件に不服である場合は、プログラムを閉じてください」です。ユーザーが [OK] をクリックすると、ウィンドウが閉じセッションを続行できます。

デフォルトの通知メッセージは、Session Recording Serverをホストするコンピューターのオペレーティングシステムの言語で表示されます。

必要な言語でカスタムの通知を作成できますが、言語ごとに作成できる通知メッセージは1つだけです。ユーザーには、ユーザーが選択したロケール設定に対応する通知メッセージが表示されます。

新しい通知メッセージを作成するには

1. Session Recording Serverをホストするコンピューターにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Server Properties] を選択します。
3. [Session Recording Server Properties] で、[通知] タブをクリックします。
4. [追加] をクリックします。
5. メッセージの言語を選択し新しいメッセージを入力します。1つの言語につき作成できるメッセージは1つです。

新しいメッセージを受け入れてアクティブにすると、[言語特有の通知メッセージ] ボックスに表示されます。



# カスタムイベントの録画を有効にする

Jan 04, 2017

Session Recordingでは、サードパーティ製のアプリケーションを使用して、イベントとして知られるカスタムデータを録画されたセッションに挿入することができます。これらのイベントは、Session Recording Playerを使用してセッションを再生するときに表示されます。これらは録画されたセッションの一部であり、録画の後に変更することはできません。

たとえば、イベントに「ユーザーがWebブラウザーを開きました」というテキストが含まれることがあります。このテキストは、録画対象のセッションでエンドユーザーがWebブラウザーを開くたびに録画に挿入されます。Session Recording Playerを使用してセッションを再生するとき、Session Recording Playerの [イベントとブックマーク] の一覧に表示されるマーカーの数から、ユーザーがWebブラウザーを開いた回数わかります。

サーバー上の録画にカスタムイベントを挿入するには

- [Session Recording Agentのプロパティ] を使用して、カスタムイベントを挿入する各サーバーで設定を有効にします。各サーバーをそれぞれ有効にする必要があります。サイト内のすべてのサーバーをまとめて有効にすることはできません。
- イベントAPIに基づくアプリケーションを開発します。このアプリケーションを各エンドユーザーのXenAppセッションで実行し、録画にデータを挿入します。

Session Recordingのインストールにはイベント記録COMアプリケーション (API) が含まれており、サードパーティ製のアプリケーションからテキストを録画に挿入することができます。Visual Basic、C++、またはC#を含む、多くのプログラミング言語でAPIを使用できます。Session RecordingイベントAPIのDLLはSession Recordingの一部としてインストールされます。APIのファイルは、%ProgramFiles%\Citrix\SessionRecording\Agent\Bin\Interop.UserApi.dllです。

サーバーでカスタムイベントの記録を有効にするには

1. Session Recording Agentがインストールされているサーバーにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Agentのプロパティ] を選択します。
3. [Session Recording Agentのプロパティ] で、[録画] タブをクリックします。
4. [カスタムイベントの記録] で [このXenAppサーバーでサードパーティ製アプリケーションによるカスタムデータの記録を許可する] チェックボックスをオンにします。

# ライブセッションの再生を有効または無効にする

Jan 04, 2017

Session Recording Playerを使用して、セッションの録画後または録画中にセッションを再生できます。録画中のセッションの再生は現在実行されている操作の表示と似ていますが、実際には1、2秒の遅れが生じます。XenAppサーバーからデータが伝達されるためです。

録画が完了していないセッションを再生するときは、次の機能は使用できません。

- 録画が完了するまでデジタル証明書を割り当てることはできません。デジタル署名が有効な場合は、ライブセッションを再生できますが、まだ録画に署名はされていません。セッションが完了して初めて、証明書を表示できます。
- 録画が完了するまで、再生データの保護は適用できません。再生データの保護が有効な場合は、ライブセッションを再生できますが、セッションが完了するまでは暗号化されません。
- 録画が完了するまで、ファイルをキャッシュできません。

デフォルトで、ライブセッションの再生は有効になっています。

1. Session Recording Serverをホストするコンピューターにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Server Properties] を選択します。
3. [Session Recording Server Properties] で、[再生] タブをクリックします。
4. [ライブセッションの再生を許可する] チェックボックスをオンまたはオフにします。

# 再生データの保護を有効または無効にする

Jan 04, 2017

セキュリティ上の安全のため、Session Recordingでは録画ファイルが自動的に暗号化されます。その後でファイルがダウンロードされ、Session Recording Playerで表示できるようになります。この再生データの保護機能により、録画ファイルをダウンロードしたユーザー以外のユーザーは、ファイルをコピーしたり表示したりできなくなります。ほかのワークステーションまたはユーザーアカウントでは、ファイルを再生できません。暗号化されたファイルは拡張子がICLEで、暗号化されていないファイルの拡張子はICLです。ファイルは、Session Recording Playerがインストールされているワークステーションのキャッシュにある間は暗号化されたままです。権限を持つユーザーがファイルを開いて初めて、暗号が解読されます。

HTTPSを使用して転送データを保護することをお勧めします。

再生データの保護は、デフォルトで有効になります。

1. Session Recording Serverをホストするコンピューターにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Server Properties] を選択します。
3. [Session Recording Server Properties] で、[再生] タブをクリックします。
4. [再生のためダウンロードされるセッションの録画ファイルを暗号化する]チェックボックスをオンまたはオフにします。

# デジタル署名を有効および無効にする

Jan 04, 2017

Session Recordingコンポーネントがインストールされているコンピューターに証明書をインストール済みである場合は、デジタル署名をセッションの録画に割り当てることにより、Session Recording環境のセキュリティを高めることができます。

デフォルトで、デジタル署名は無効になっています。

1. Session Recording Serverをホストするコンピューターにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Server Properties] を選択します。
3. [Session Recording Server Properties] で、[署名] タブをクリックします。
4. Session Recordingコンポーネントがインストールされているコンピューターの間で安全な通信を有効にする証明書を参照します。

1. Session Recording Serverをホストするコンピューターにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Server Properties] を選択します。
3. [Session Recording Server Properties] で、[署名] タブをクリックします。
4. [消去] をクリックします。

# 録画の格納先の指定

Jan 04, 2017

[Session Recording Server Properties] を使用して、録画の格納先とアーカイブされた録画の復元先を指定します。

注：ファイルをアーカイブまたは削除されたファイルを復元するには、icldbコマンドを使用します。

デフォルトでは、録画はSession Recording Serverをホストするコンピューターの<drive>:\SessionRecordingsフォルダーに格納されます。録画を格納するフォルダーを変更したり、複数のボリューム間で負荷分散をするため、または追加の容量を活用するために、フォルダーを追加したりできます。複数のフォルダーが一覧にある場合は、録画がフォルダー間で負荷分散されていることを示します。1つのフォルダーを複数回追加できます。負荷分散は各フォルダーを循環して行われます。

1. Session Recording Serverをホストするコンピューターにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Server Properties] を選択します。
3. [Session Recording Server Properties] で、[格納場所] タブをクリックします。
4. [ファイル格納フォルダー] ボックスの一覧を使用して、録画を格納するフォルダーを管理します。

ローカルドライブ、SANボリューム、またはUNCネットワークパスで指定する場所にファイル格納フォルダーを作成できます。マップされたネットワークドライブのドライブ文字はサポートされません。Session Recordingでは、NAS (Network-Attached Storage : ネットワークアタッチトストレージ) を使用しないでください。ネットワークドライブへの録画データの書き込みに関連して、パフォーマンスおよびセキュリティ上の重大な問題が起きる可能性があります。

デフォルトでは、アーカイブされた録画はSession Recording Serverをホストするコンピューターの<drive>\SessionRecordingsRestoreフォルダーに復元されます。アーカイブされた録画を復元するフォルダーは変更できません。

1. Session Recording Serverをホストするコンピューターにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Server Properties] を選択します。
3. [Session Recording Server Properties] で、[格納場所] タブをクリックします。
4. [アーカイブ済みファイルの復元フォルダー] ボックスにアーカイブ済みファイルを復元するフォルダーを入力します。

# 録画ファイルのサイズの指定

Jan 04, 2017

録画ファイルのサイズが大きくなるにつれて、ダウンロードに時間がかかり、再生中にシークスライダーを使用して再生箇所を変更するとき反応が遅くなるようになります。ファイルサイズを制御するにはファイルのしきい値を指定します。録画ファイルがこの限界に達すると、Session Recordingによってファイルが閉じられ、録画を続行するために新しいファイルが開かれます。この操作をロールオーバーと呼びます。

ロールオーバーのため、2つのしきい値を指定できます。

- **ファイルサイズ**。ファイルがMB単位で指定された大きさに達すると、ファイルが閉じられ、新しいファイルが開かれます。デフォルトでは、ファイルが50MBに達するとロールオーバーが起こります。ただし、10MBから1GBの範囲で上限を指定できます。
- **時間**。セッションが時間単位で指定された時間録画されると、ファイルが閉じられ、新しいファイルが開かれます。デフォルトでは、12時間録画するとロールオーバーが起こります。ただし、1時間から24時間の範囲で上限を指定できます。

Session Recordingにより両方のボックスの値が確認され、ロールオーバーを実行するタイミングを決定するイベントとして、どちらのしきい値を先に超過するかが判断されます。たとえば、ファイルサイズに17MB、時間に6時間を指定し、録画ファイルのサイズが3時間で17MBに達したとします。この場合、17MBのファイルサイズに対応してロールオーバー処理が起動し、ファイルが閉じられ、新しいファイルが開かれます。

多くの小さなファイルが作成されないように、ファイルサイズに指定された値にかかわらず、少なくとも1時間（指定できる最小の値）が経過するまでロールオーバーは起こりません。この規則の例外は、ファイルサイズが1GBを超えた場合です。

1. Session Recording Serverをホストするコンピューターにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Server Properties] を選択します。
3. [Session Recording Server Properties] で、[ロールオーバー] タブをクリックします。
4. 10から1024の間の整数を入力して、ファイルサイズの上限をMB単位で指定します。
5. 1から24の間の整数を入力して、録画時間の上限を時間単位で指定します。

# 録画の表示

Jan 04, 2017

Session Recording Playerを使用して、録画したXenAppまたはXenDesktopセッションを表示、検索、およびブックマークします。

ライブセッションの再生機能を有効にしてセッションを録画する場合は、完了したセッションはもちろん、進行中のセッションも数秒遅れで表示できます。

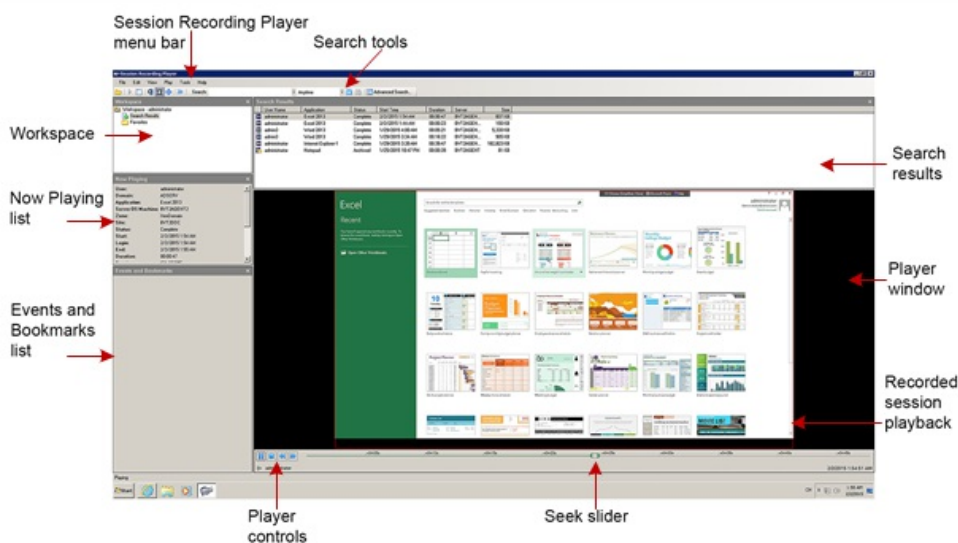
Session Recording管理者が設定する時間やファイルサイズの上限を超えるセッションは、複数のセッションファイルに分けて表示されます。

注：Session Recording管理者が、録画されたサーバーOSマシンセッションへのアクセス権をユーザーに付与する必要があります。セッションを表示できない場合は、Session Recording管理者に連絡してください。

通常、Session Recording管理者がSession Recording Playerをインストールするとき、Session Recording PlayerとSession Recording Serverの間の接続も設定します。この接続が未設定の場合は、初めてファイルを検索するときに設定のダイアログボックスが開きます。設定に必要な情報についてはSession Recording管理者に問い合わせてください。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。  
Session Recording Playerが表示されます。

この図はSession Recording Playerとその主要要素を示します。これらの要素の機能について、以下で説明します。



Session Recording Playerには、表示するかどうかを切り換えるためのウィンドウ要素があります。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. Session Recording Playerのメニューバーで、[表示] を選択します。
4. 表示する要素を選択します。選択するとすぐにその要素が表示されます。チェックマークはその要素が選択されていることを示します。

Session Recording管理者がSession Recording Playerから複数のSession Recording Serverに接続できるように設定した場合は、Session Recording Playerが接続するSession Recording Serverを選択できます。Session Recording Playerから同時に複数のSession Recording Serverに接続することはできません。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. Session Recording Playerのメニューバーで、[ツール]、[オプション]、[接続] の順に選択します。
4. 接続するSession Recording Serverを選択します。



# 録画の再生

Jan 04, 2017

Session Recording Playerでセッションの録画を開くには、次の3つの方法があります。

- Session Recording Playerを使用して検索を実行する。検索条件に一致するセッションの録画が、検索結果の領域に表示されます。
- ローカルディスクドライブまたは共有ドライブ上のセッションの録画ファイルに直接アクセスする。
- お気に入りフォルダーからセッションの録画ファイルにアクセスする。

デジタル署名なしで録画されたファイルを開くと、その元ファイルと整合性が検証されていないという警告が表示されます。ファイルの整合性について確信がある場合は、警告のポップアップウィンドウで [[はい]] をクリックしてファイルを開きます。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. 検索を実行します。
4. 検索結果の領域が表示されていない場合は、[ワークスペース] ペインの[検索結果] を選択します。
5. 検索結果の領域で、再生するセッションを選択します。
6. 次のいずれかを実行します。
  - セッションをダブルクリックします。
  - セッションを右クリックして [再生] を選択します。
  - Session Recording Playerのメニューバーで、[再生] の [再生] を選択します。

セッションの録画ファイルの名前は、冒頭が付く一意な英数字のファイルIDで、拡張子は、再生データの保護機能が無効な録画の場合は.icl、有効な録画の場合は.icleになります。セッションの録画ファイルが保存されるディレクトリ構造には、セッションを録画した日付が反映されます。たとえば、2014年12月22日に録画されたセッションのファイルは、フォルダーパス 2014\12\22に保存されます。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. 次のいずれかを実行します。
  - メニューバーで、[ファイル] の [開く] を選択し、ファイルを参照します。
  - Windowsのエクスプローラーを使用してファイルを表示し、ファイルをプレーヤーウィンドウにドラッグします。
  - Windowsのエクスプローラーを使用してファイルを表示し、ダブルクリックします。
  - [ワークスペース] ペインにお気に入りを作成した場合は[お気に入り] を選択して、検索結果の領域から開く場合と同様に、お気に入りの領域からファイルを開きます。

お気に入りフォルダーを作成して、頻繁に表示する録画にすばやくアクセスすることができます。お気に入りフォルダーによって、ワークステーションまたはネットワークドライブに格納されているセッションの録画ファイルが参照されます。これらのファイルはほかのワークステーションとの間でインポートとエクスポートをして、ほかのSession Recording Playerのユーザーと共有できます。

注：Session Recording Playerへのアクセス権を持つユーザーのみが、お気に入りフォルダーに関連付けられているセッション

ンの録画ファイルをダウンロードできます。アクセス権については、Session Recording Player管理者にお問い合わせください。

お気に入りサブフォルダーを作成するには

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. Session Recording Playerの [ワークスペース] ペインで [お気に入り] フォルダーを選択します。
4. メニューバーで [ファイル]、[フォルダー]、[フォルダーの作成] の順に選択します。 [お気に入り] フォルダーの下の階層に新しいフォルダーが表示されます。
5. フォルダー名を入力し、Enterキーを押すか任意の場所をクリックして新しい名前を受け入れます。

[ファイル] の [フォルダー] を選択すると表示されるほかのオプションを使用して、フォルダーの削除、名前の変更、移動、コピー、インポート、およびエクスポートを行います。

# セッションの録画の検索

Jan 04, 2017

Session Recording Playerでは、クイック検索を実行することも、高度な選択を実行して検索に適用するオプションを指定することもできます。検索結果はSession Recording Playerの検索結果の領域に表示されます。

注：使用できる録画されたセッションを1回の検索で表示できるセッション数の上限まですべて表示するには、検索パラメーターを指定せずに検索を実行します。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. 次の方法で検索条件を定義します。
  - [検索] ボックスに検索条件を入力します。次の補助機能を使用できます。
  - [検索] ラベルの上にマウスポインターを移動すると、入力できるパラメーターの一覧が表示されます。
  - [検索] ボックス右側の矢印をクリックすると、過去に使用した検索文字列が最新の64件まで表示されます。
  - [検索] ボックス右側のドロップダウンリストを使用して、セッションが録画された期間を指定できます。
4. ドロップダウンリスト右側の双眼鏡のアイコンをクリックして、検索を開始します。

ヒント：高度な検索では、150,000個を超えるエンティティが含まれる結果が返されるのに最大20秒かかる場合があります。日付範囲やユーザーなどのより厳密な検索条件を使用して、結果の数を減らすことをお勧めします。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. [Session Recording Player] ウィンドウで、ツールバーの [高度な検索] をクリックするか、メニューバーで [ツール] の [高度な検索] を選択します。
4. [高度な検索] ダイアログボックスのタブで検索条件を定義します。
  - [共通] タブでは、ドメインまたはアカウントの認証先、サイト、グループ、サーバーOSマシン、アプリケーション、またはファイルIDを使用して検索できます。
  - [日付/時刻] タブでは、日付、曜日、および時刻を使用して検索できます。
  - [イベント] タブでは、Session Recording管理者がセッションに挿入したカスタムイベントを使用して検索できます。
  - [そのほか] タブでは、セッション名、クライアント名、クライアントアドレス、および録画時間を使用して検索できます。このタブでは、表示される検索結果数の上限およびアーカイブ済みのファイルを検索に含めるかどうかも指定できます。

検索条件を指定するにつれて、作成しているクエリがダイアログボックス下部のペインに表示されます。

5. [検索] をクリックして検索を開始します。

ヒント：高度な検索のクエリは、保存しておいて後で取得することができます。[高度な検索] ダイアログボックスの [保存] をクリックして、現在のクエリを保存します。[高度な検索] ダイアログボックスの [開く] をクリックして、保存したクエリを取得します。保存したクエリファイルの拡張子は.isqです。

Session Recording Playerの検索オプションにより、表示される検索結果数の上限およびアーカイブ済みのファイルを検索に含めるかどうか指定できます。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. メニューバーで、[ツール]、[オプション]、[検索] の順に選択します。

4. [検索結果の表示件数の上限] ボックスに、表示する検索結果数を入力します。最大で500件の検索結果を表示できません。
5. アーカイブ済みのファイルを検索に含めるかどうかを設定するには、[アーカイブ済みファイルを含める] チェックボックスをオンまたはオフにします。

# セッションの録画の再生

Jan 04, 2017

Session Recording Playerでセッションの録画を開いた後は、次の方法で録画されたセッション内を移動できます。






- プレーヤーウィンドウのボタンを使用して、再生、停止、一時停止、および再生速度の変更を行います。
- シークスライダーを使用して、前後に移動します。

マーカーが録画に挿入されている場合、およびセッションの録画にカスタムイベントが含まれている場合は、マーカーおよびイベントのポイントに移動することによって、録画されたセッション内を移動することもできます。

注：

- セッションの録画の再生時に、マウスポインターが2つ表示される場合があります。この問題は、ユーザーがInternet Explorerを使用中に、Internet Explorerにより自動的に縮小表示されたイメージをユーザーがクリックすると発生します。セッション内では1つのマウスポインターしか表示されませんが、セッションの録画の再生時にのみ2つ目のマウスポインターが表示されます。
- このバージョンのSession Recordingは、XenAppのSpeedScreenマルチメディアアクセラレーション機能や [Flash品質の調整] ポリシー設定をサポートしません。この機能が有効の場合、再生画面が黒く表示されます。
- HDX RealTime Optimization Pack for Microsoft Lyncを使用している場合、Session RecordingでLync Webカメラの映像を録画することはできません。

プレーヤーウィンドウのボタンを使用するか、メニューバーの[再生] の下のメニューアイテムを選択して、セッションの録画を操作します。プレーヤーウィンドウのボタンまたはメニューアイテムを使用して次の操作を実行します。

|                                                                                     |                                                      |
|-------------------------------------------------------------------------------------|------------------------------------------------------|
|  | 選択したセッションファイルを再生します。                                 |
|  | 再生を一時停止します。                                          |
|  | 再生を停止します。 [停止] 、 [再生] の順にクリックすると、ファイルの冒頭から再生が開始されます。 |
|  | 現在の再生速度の半分に速度を変更します。最低で標準の4分の1にまで速度を下げます。            |
|  | 現在の再生速度の2倍に速度を変更します。最高で標準の32倍にまで速度を上げます。             |

プレーヤーウィンドウ下部のシークスライダーを使用して、セッションの録画内の別の位置にジャンプします。シークスライダーを録画内の表示したいポイントまでドラッグすることも、スライダーバーの任意のポイントをクリックして移動することもできます。

また、次のキーボードキーを使用してシークスライダーを制御できます。

| キー： | シーク操作： |
|-----|--------|
|     |        |

|                                |              |
|--------------------------------|--------------|
| Home<br>キー                     | 冒頭へシークします。   |
| End                            | 末尾へシークします。   |
| 右矢印                            | 5秒先へシークします。  |
| 左矢印                            | 5秒前へシークします。  |
| マウスホイールを1目盛り手前に動かす             | 15秒先へシークします。 |
| マウスホイールを1目盛り奥に動かす              | 15秒前へシークします。 |
| Ctrl+ 右矢印                      | 30秒先へシークします。 |
| Ctrl+ 左矢印                      | 30秒前へシークします。 |
| Page Down                      | 1分先へシークします。  |
| Page Up                        | 1分前へシークします。  |
| Ctrlキーを押しながらマウスホイールを1目盛り手前に動かす | 90秒先へシークします。 |
| Ctrlキーを押しながらマウスホイールを1目盛り奥に動かす  | 90秒前へシークします。 |
| Ctrl+ Page Down                | 6分先へシークします。  |
| Ctrl+ Page Up                  | 6分前へシークします。  |

注：シークスライダーの速度を調整するには、メニューバーで、[ツール]、[オプション]、[プレーヤー]の順に選択し、スライダーをドラッグしてシークの応答速度を変更します。応答速度を上げると、より多くのメモリが消費されます。録画のサイズやマシンのハードウェアによって、応答速度が低下する場合があります。

Session Recording Playerを設定して、標準の4分の1倍速から32倍速までの速度で、セッションの録画を再生できます。速度は指数的に増加します。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. メニューバーで、[再生] の [再生速度] を選択します。
4. 速度オプションを選択します。

すぐに速度が調節されます。速度を示す数字がプレーヤーウィンドウのボタンの下に表示されます。この速度を示す緑色の:

キストは、プレーヤーウィンドウにも短時間表示されます。

高速レビューモードを使用すると、録画されたセッション内で操作のない部分の再生を省略することができます。この設定により再生時間を短くできます。ただし、アニメーションを用いたマウスポインター、点滅するカーソル、秒針付きの時計など、動画による連続処理の再生は省略できません。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. Session Recording Playerのメニューバーで、[再生] の [高速レビューモード] を選択します。

オプションがオンまたはオフに切り替わります。このオプションを選択するたびに、その状態がプレーヤーウィンドウに短時間表示されます。

# イベントとブックマークの使用

Jan 04, 2017

イベントとブックマークを使用して、録画されたセッション内を簡単に移動できます。

イベントは、イベントAPIおよびサードパーティ製のアプリケーションを使用して、セッションの録画中に挿入されます。イベントはセッションファイルの一部として保存されます。Session Recording Playerを使用して削除または変更することはできません。

ブックマークは、録画されたセッションにSession Recording Playerを使用して挿入するマーカーです。ブックマークは削除するまでは録画されたセッションに関連付けられますが、セッションファイルの一部としては保存されません。各ブックマークのデフォルトのラベルテキストは「ブックマーク」ですが、最長128文字までの任意のコメントテキストに変更できません。

イベントとブックマークはプレーヤーウィンドウの下部に丸印として表示されます。イベントの丸印は黄色で、ブックマークの丸印は青です。これらの丸印にポインターを合わせると、関連付けられているテキストラベルが表示されます。イベントとブックマークは、Session Recording Playerのイベントとブックマーク一覧にも表示できます。そのテキストラベルと録画されたセッションでの時刻と共に、時系列で一覧に表示されます。

イベントとブックマークを使用して、録画されたセッション内を簡単に移動できます。イベントまたはブックマークに移動することにより、それらが挿入されているポイントまでを省略して、録画されたセッション内を移動できます。

イベントとブックマークの一覧には、現在再生中の録画されたセッションに挿入されているイベントとブックマークが表示されます。イベントのみ、ブックマークのみ、または両方を表示できます。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. イベントとブックマークの一覧にマウスポインターを移動し、右クリックしてメニューを表示します。
4. [イベントのみ表示]、[ブックマークのみ表示]、または[すべて表示] を選択します。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. ブックマークを追加する、録画されたセッションの再生を開始します。
4. ブックマークを挿入する位置までシークスライダーを動かします。
5. プレーヤーウィンドウ内にマウスポインターを移動し、右クリックしてメニューを表示します。
6. 次の方法で、デフォルトのラベル「ブックマーク」でブックマークを追加するか、コメントを作成します。
  - デフォルトのラベル「ブックマーク」でブックマークを追加するには、[ブックマークを追加] を選択します。
  - 内容のわかるテキストラベルを作成してブックマークを追加するには、[コメントを追加] を選択します。ブックマークに割り当てるテキストラベルを最長128文字で入力します。[OK] をクリックします。

ブックマークを作成した後でコメントを追加したり、コメントを変更したりできます。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. ブックマークを含む録画されたセッションの再生を開始します。
4. イベントとブックマークの一覧にブックマークが表示されていることを確認します。



5. イベントとブックマークの一覧でブックマークを選択し、右クリックしてメニューを表示します。
6. [コメントを編集] を選択します。
7. 表示されるダイアログボックスに新しいコメントを入力して[OK] をクリックします。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. ブックマークを含む録画されたセッションの再生を開始します。
4. イベントとブックマークの一覧にブックマークが表示されていることを確認します。
5. イベントとブックマークの一覧でブックマークを選択し、右クリックしてメニューを表示します。
6. [削除] を選択します。

イベントまたはブックマークに移動すると、それらが挿入されているポイントまでを省略して、録画されたセッション内を移動できます。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. イベントまたはブックマークを含む録画されたセッションの再生を開始します。
4. 次の方法で、イベントまたはブックマークに移動します。
  - プレーヤーウィンドウの下部でイベントまたはブックマークを表す丸印をクリックします。
  - イベントとブックマークの一覧で、イベントまたはブックマークをダブルクリックします。次のイベントまたはブックマークに移動するには、一覧から任意のイベントまたはブックマークを選択して右クリックし、[ブックマークへシーク] を選択します。

# 再生の表示形式の変更

Jan 04, 2017

オプションを使用して、プレーヤーウィンドウにセッションの録画を表示する形式を変更できます。セッションの録画をプレーヤーウィンドウまたは元のセッションのサイズに合わせて表示したり、全画面で再生したり、プレーヤーウィンドウを独立ウィンドウで表示したり、セッションの録画の周りに赤い枠を表示して、セッションをプレーヤーウィンドウの背景と区別しやすくしたりできます。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. Session Recording Playerのメニューバーで、[表示] の [全画面] を選択します。
4. 元のサイズに戻すには、EscキーまたはF11キーを押します。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. Session Recording Playerのメニューバーで、[表示] の [独立ウィンドウ] を選択します。プレーヤーウィンドウを含む新しいウィンドウが開きます。ウィンドウをドラッグしてサイズを変更できます。
4. プレーヤーウィンドウをメインウィンドウに埋め込むには、メニューバーで[表示] の [独立ウィンドウ] を選択するか、F10キーを押します。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. Session Recording Playerのメニューバーで、[再生]、[表示モード]、[ウィンドウに合わせる] の順に選択します。
  - [ウィンドウに合わせる (高速描画)] を選択すると、十分な画質を維持しながら画像を縮小します。高画質オプションを使用する場合より描画が高速で行われますが、画像とテキストの明晰さは低下します。高画質モードでパフォーマンスに問題が生じる場合は、このオプションを使用します。
  - [ウィンドウに合わせる (高画質)] を選択すると、明晰な画像とテキストを維持しながら画像を縮小します。このオプションを使用すると、高速描画オプションの場合より描画速度が遅くなる場合があります。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. Session Recording Playerのメニューバーで、[再生]、[表示モード]、[セッションに合わせる] の順に選択します。ポインターが手のひらの形に変化し、画面全体を表す小さなイメージがプレーヤーウィンドウの右上に表示されます。
4. 画面をドラッグします。小さなイメージに現在表示されている箇所が表示されます。
5. スクロールを停止するには、ウィンドウに合わせるオプションのどちらかを選択します。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. Session Recording Playerのメニューバーで、[ツール]、[オプション]、[プレーヤー] の順に選択します。
4. [セッションの録画の周りに枠線を表示する] チェックボックスをオンにします。

ヒント： [セッションの録画の周りに枠線を表示する] チェックボックスがオフの場合は、マウスポインターがプレーヤーウィンドウ内にあるときにマウスボタンを押したままにすると、一時的に赤い枠線が表示されます。

# セッションの録画ファイルのキャッシュ

Jan 04, 2017

セッションの録画ファイルを開くたびに、録画が格納されている場所からファイルがダウンロードされます。同じファイルを頻繁にダウンロードする場合は、ファイルをワークステーションにキャッシュすることでダウンロード時間を節約できます。ワークステーションにキャッシュされるファイルは次のフォルダーに格納されます。

<userprofile>AppData\Local\Citrix\SessionRecording\Player\Cache

キャッシュに割り当てるディスク容量を指定できます。指定した容量まで録画ファイルが蓄積されると、最も古く使用されていない録画が削除され、新しい録画のための空き領域が作成されます。ディスク領域を解放するために、いつでもキャッシュを空にすることができます。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. Session Recording Playerのメニューバーで、[ツール]、[オプション]、[キャッシュ]の順に選択します。
4. [ダウンロードしたファイルをローカルコンピューターにキャッシュする]チェックボックスをオンにします。
5. キャッシュに使用されるディスク容量を制限するには、[使用するディスク容量を制限する]チェックボックスをオンにして、使用する容量をメガバイト単位で指定します。
6. [OK] をクリックします。

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
3. Session Recording Playerのメニューバーで、[ツール]、[オプション]、[キャッシュ]の順に選択します。
4. [ダウンロードしたファイルをローカルコンピューターにキャッシュする]チェックボックスをオンにします。
5. Session Recording Playerで、[ツール]、[オプション]、[キャッシュ]の順に選択します。
6. [キャッシュの削除] をクリックし、次に [OK] をクリックして操作を確定します。

# Session Recordingのトラブルシューティング

Jan 04, 2017

このトラブルシューティング情報には、Session Recordingコンポーネントのインストール中とインストール後に発生する可能性のある、次のような問題に対する解決策が含まれています。

- コンポーネントで相互接続できない。
- セッションを録画できない。
- Session Recording PlayerまたはSession Recording Policy Consoleの問題。
- 通信プロトコルに関連して問題が発生する。

Session Recording Agentから接続できないとき、「Session Recording Brokerにポールメッセージを送信しています - この処理の実行中に例外が見つかりました。」というイベントメッセージが、例外のテキストと共にログに記録されます。例外のテキストには接続に失敗した原因が記載されます。次のような状況で、データの移行が必要です。

- 「基になる接続が閉じられました。SSL/TLSの安全なチャンネルを確立できませんでした。」この例外は、Session Recording Serverで使用している証明書を署名したCAが、Session Recording Agentが動作するサーバーに信頼されていないか、Session Recording Agentが動作するサーバーにCA証明書がインストールされていないことを示します。または、証明書の有効期限が切れているか失効している可能性があります。

解決策：正しいCA証明書がSession Recording Agentをホストするサーバーにインストールされているか、信頼されているCAを使用していることを確かめます。

- 「リモートサーバーがエラーを返しました。HTTP 403 (アクセス不可)。」これは、HTTP (セキュリティで保護されていないプロトコル) を使用して接続しようとしたときに表示される、標準のHTTPSエラーです。Session Recording Serverをホストするコンピューターでセキュリティで保護された接続のみを受け付けるため、接続が拒否されています。

解決策：[Session Recording Agentのプロパティ] を使用してSession Recording BrokerのプロトコルをHTTPSに変更します。

「レコードポリシークエリの検証中に、Session Recording Brokerが不明なエラーを返しました。エラーコード5 (アクセスが拒否されました)。詳しくは、Session Recording Serverでイベントログを参照してください。」このエラーは、セッションが開始され録画ポリシーの評価要求が送信されると発生します。このエラーは、Session Recording承認コンソールの役割であるポリシークエリの役割から、デフォルトのメンバーであるAuthenticated Usersグループが削除された結果発生します。

解決策：Authenticated Usersグループをこの役割に再追加するか、各Session Recording Agentをホストする各サーバーをポリシークエリの役割に追加します。

「基になる接続が閉じられました。維持される必要があった接続が、サーバーによって切断されました。」このエラーは、Session Recording Serverが停止しているか、要求を受け付けられないことを示します。IISがオフラインまたは再起動中であることが原因か、サーバー全体がオフラインである可能性があります。

解決策：Session Recording Serverが開始されていること、サーバーでIISが実行中であること、およびサーバーがネットワークに接続していることを確かめます。

Session Recording ServerからSession Recordingデータベースに接続できないとき、次のいずれかのメッセージが表示されることがあります。

イベントソース：

「Citrix Session Recordingストレージマネージャーの説明: データベース接続を確立しています - この処理の実行中に例外が見つかりました。」このエラーは、Session Recording Serverをホストするコンピューターのイベントビューアーのアプリケーションログに記録されます。

「Session Recording Serverに接続できません。Session Recording Serverが実行中か確認してください。」このエラーメッセージはSession Recording Policy Consoleを起動するときに表示されます。

解決策：

- Microsoft SQL Server 2008 R2、Microsoft SQL Server 2012、またはMicrosoft SQL Server 2014のExpress Editionがスタンドアロンサーバーにインストールされていますが、Session Recordingのサービスまたは設定が正しく設定されていません。サーバーでTCP/IPプロトコルを有効にしてSQL Server Browserサービスを実行する必要があります。これらの設定を有効にする方法については、Microsoftのドキュメントを参照してください。
- Session Recording管理ツールのインストール中に、サーバーとデータベースについて誤った情報が指定されています。Session Recordingデータベースをアンインストールし、正しい情報を指定して再インストールします。
- Session Recordingデータベースサーバーが停止しています。サーバーに接続できることを確かめます。
- Session Recording ServerまたはSession Recordingデータベースサーバーをホストするコンピューターで、もう一方のFQDNまたはNetBIOS名を解決できません。pingコマンドを使用して、名前を解決できることを確認します。

ユーザー 'NT\_AUTHORITY\ANONYMOUS LOGON' ログオン失敗。このエラーメッセージは、サービスのログオンアカウントが誤って、\administratorになっていることを意味します。

解決策：ローカルシステムユーザーとしてサービスを再起動し、SQLサービスを再起動します。

アプリケーションセッションが録画されない場合は、まずSession Recording AgentとSession Recording Serverが動作するサーバーOSマシンのイベントビューアーで、アプリケーションイベントログを確認します。このログには有益な診断情報が含まれている可能性があります。

セッションが録画されない場合、次の問題が原因である可能性があります。

- **コンポーネント間の接続と証明書。** Session Recordingコンポーネントの間で通信ができない場合は、セッションの録画に失敗する可能性があります。録画の問題のトラブルシューティングをするには、すべてのコンポーネントが適切に設定されていて正しいコンピューターを参照していることと、すべての証明書が有効で適切にインストールされていることを確認します。
- **Active Directoryドメインでない環境。** Session RecordingはMicrosoft Active Directoryドメインの環境で動作するように設計されています。Active Directory環境で運用していない場合は、録画で問題が発生する可能性があります。Session Recordingのすべてのコンポーネントは、必ずActive Directoryドメインに参加しているコンピューターで実行します。
- **セッション共有がアクティブなポリシーと競合している。** Session Recordingでは、アクティブなポリシーとエンドユーザーが最初に開いた公開アプリケーションを照合します。同じセッション上で次のアプリケーションを開くと、最初のアプリケーションに対して有効なポリシーが、次のアプリケーションにも適用されます。セッション共有がアクティブなポリシーと競合することを防ぐには、競合するアプリケーションを別のサーバーOSマシンで公開します。
- **録画が有効になっていない。** サーバーOSマシンにSession Recording Agentをインストールすると、そのサーバーではデフォルトで録画が有効になります。録画を許可するアクティブな録画ポリシーを設定するまでは、録画はされません。
- **アクティブな録画ポリシーによって録画が許可されない。** セッションを録画するには、特定のユーザー、サーバー、または公開アプリケーションを対象に、アクティブな録画ポリシーでセッションの録画が許可されている必要があります。
- **Session Recordingサービスが実行されていない。** セッションを録画するには、サーバーOSマシンでSession Recording Agentサービスが実行されており、Session Recording ServerをホストするコンピューターでSession Recording Storage

Managerサービスが実行されている必要があります。

- MSMQが設定されていない。Session Recording Agentが動作するサーバーとSession Recording ServerをホストするコンピュータでMSMQが適切に設定されていない場合は、録画の問題が起きる可能性があります。

Session Recording Playerで録画を再生できないときは、次のエラーメッセージが表示される可能性があります。

「セッションの録画ファイルをダウンロードできませんでした。ライブセッションの再生は許可されていません。サーバーがこの機能を許可しない設定になっています。」このエラーは、サーバーがこの操作を許可しないように設定されていることを示します。

解決策： [Session Recording Serverのプロパティ] ダイアログボックスで [再生] タブをクリックし、 [ライブセッションの再生を許可する] チェックボックスをオンにします。

Session Recording Playerを使用して表示しているときに録画が破損したり不完全だったりする場合は、Session Recording Agentのイベントログにも警告が記録されている可能性があります。

イベントソース： Citrix Session Recording Storage Manager

説明： 次のファイルの記録中にデータが失われました：

これは通常、Machine Creation Services (MCS) またはProvisioning Servicesで、構成済みのマスターイメージとインストール済みのMicrosoft Message Queuing (MSMQ) を使用してVDAを作成する場合に発生します。この条件ではVDAのQMIDはMSMQと同じになります。

解決策： 各VDAに固有のQMIDを作成します。回避策が「[既知の問題](#)」に記載されています。

Session RecordingデータベースまたはSession Recording Serverのインストール中に、データベースインスタンス名が正しくても「データベース接続テストに失敗しました。正しいデータベースインスタンス名を入力してください。」というエラーメッセージと共にテスト接続に失敗します。

解決策： 権限の制限による失敗を修正するため、現在のユーザーがSQL Serverに対してpublicのロールを持っていることを確認してください。

# コンポーネント間の接続の確認

Jan 04, 2017

Session Recordingのセットアップ中にコンポーネント間の接続に成功しないことがあります。すべてのコンポーネントがSession Recording Server (Broker) と通信を行います。デフォルトでは、IISのコンポーネントであるブローカーのセキュリティは、IISの既定のWebサイトの証明書を使用して保護されます。あるコンポーネントからSession Recording Serverに接続できないときは、ほかのコンポーネントから接続を試行しても失敗することがあります。

Session Recording AgentとSession Recording Server (ストレージマネージャーとブローカー) の接続エラーは、Session Recording Serverをホストするコンピューターの、イベントビューアーのアプリケーションログに記録されます。Session Recording Policy ConsoleとSession Recording Playerでは、接続に失敗したときに画面にエラーメッセージが表示されます。

1. Session Recording Agentがインストールされているサーバーにログオンします。
2. [スタート] ボタンをクリックし、[Session Recording Agentのプロパティ] を選択します。
3. [Session Recording Server Properties] で、[接続] をクリックします。
4. [Session Recording Server] の値が、Session Recording Serverをホストするコンピューターの正しい名前であることを確認します。
5. [Session Recording Server] の値として入力されているサーバーにサーバーOSマシンから通信できることを確かめます。

注：イベントビューアーのアプリケーションログにエラーや警告が記録されていないか確認します。

注意：レジストリエディターの使用を誤ると、深刻な問題が発生する可能性があり、Windowsの再インストールが必要になる場合もあります。レジストリエディターの誤用による障害に対して、Citrixでは一切責任を負いません。レジストリエディターは、お客様の責任と判断の範囲でご使用ください。

1. Session Recording Serverをホストするコンピューターにログオンします。
2. レジストリエディターを開きます。
3. HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\SmartAuditor\Serverを表示します。
4. SmAudDatabaseInstanceの値で、SQL ServerのインスタンスにインストールしたSession Recordingデータベースが正しく参照されていることを確かめます。

1. SQL管理ツールを使用して、インストールしたSession Recordingデータベースを含むSQLインスタンスを開きます。
2. Session Recordingデータベースのセキュリティ許可を開きます。
3. Session Recordingコンピューターアカウントにデータベースへのアクセス許可が与えられていることを確かめます。たとえば、Session Recording Serverをホストするコンピューターの名前がMISドメインのSsRecSrvである場合、データベースにコンピューターアカウントとしてMIS\SsRecSrvを指定する必要があります。この値はSession Recordingデータベースのインストール中に設定します。

Session Recordingコンポーネントで通信に問題が起きたときは、WebブラウザでSession Recording BrokerのWebページにアクセスしてSession Recording ServerのIISサイトへの接続をテストすると、問題の原因が、プロトコルの誤設定なのか、証明書の問題なのか、Session Recording Brokerの問題なのかを判断するのに役立ちます。

## Session Recording AgentのIISの接続を確かめるには



1. Session Recording Agentがインストールされているサーバーにログオンします。
2. Webブラウザを起動して次のアドレスを入力します。
  - HTTPSで接続する場合：https://<servername>/SessionRecordingBroker/RecordPolicy.rem?wsdl（ここで、<servername>はSession Recording Serverをホストするコンピューターの名前です）
  - HTTPで接続する場合：http://<servername>/SessionRecordingBroker/RecordPolicy.rem?wsdl（ここで、<servername>はSession Recording Serverをホストするコンピューターの名前です）
3. NTLM（NT LAN Manager）認証の資格情報の入力を求められた場合は、ドメイン管理者のアカウントでログオンします。

WebブラウザにXMLドキュメントが表示された場合は、設定したプロトコルを使用して、Session Recording Agentが動作するコンピューターがSession Recording Serverをホストするコンピューターと接続していることが確かめられたことになります。

## Session Recording PlayerのIISの接続を確かめるには

1. Session Recording Playerがインストールされているワークステーションにログオンします。
2. Webブラウザを起動して次のアドレスを入力します。
  - HTTPSで接続する場合：https://<servername>/SessionRecordingBroker/Player.rem?wsdl（ここで、<servername>はSession Recording Serverをホストするコンピューターの名前です）
  - HTTPで接続する場合：http://<servername>/SessionRecordingBroker/Player.rem?wsdl（ここで、<servername>はSession Recording Serverをホストするコンピューターの名前です）
3. NTLM（NT LAN Manager）認証の資格情報の入力を求められた場合は、ドメイン管理者のアカウントでログオンします。

WebブラウザにXMLドキュメントが表示された場合は、設定したプロトコルを使用して、Session Recording Playerが動作するコンピューターがSession Recording Serverをホストするコンピューターと接続していることが確かめられたことになります。

## Session Recording Policy ConsoleのIISの接続を確かめるには

1. Session Recording Policy Consoleがインストールされているサーバーにログオンします。
2. Webブラウザを起動して次のアドレスを入力します。
  - HTTPSで接続する場合：https://<servername>/SessionRecordingBroker/PolicyAdministration.rem?wsdl（ここで、<servername>はSession Recording Serverをホストするコンピューターの名前です）
  - HTTPで接続する場合：http://<servername>/SessionRecordingBroker/PolicyAdministration.rem?wsdl（ここで、<servername>はSession Recording Serverをホストするコンピューターの名前です）
3. NTLM（NT LAN Manager）認証の資格情報の入力を求められた場合は、ドメイン管理者のアカウントでログオンします。

WebブラウザにXMLドキュメントが表示された場合は、設定したプロトコルを使用して、Session Recording Policy Consoleが動作するコンピューターがSession Recording Serverをホストするコンピューターと接続していることが確かめられたことになります。

通信プロトコルにHTTPSを使用する場合は、Session Recording Serverをホストするコンピューターにサーバー証明書を設定する必要があります。すべてのコンポーネントのSession Recording Serverへの接続に、ルートCA（Certificate Authority：証明機関）が必要です。この証明書をインストールしないと、コンポーネント間の通信は失敗します。

IISの接続をテストするときのように、Session Recording BrokerのWebページにアクセスすることによって、証明書をテストすることができます。各コンポーネントのXMLページにアクセスできる場合は、証明書は正しく設定されています。

ここでは、接続エラーの原因になる、証明書によくある問題について説明します。

- **無効な証明書または証明書の不足**：Session Recording Agentが動作するサーバーにサーバー証明書を信頼するためのルート証明書がインストールされていない場合は、HTTPSを介してSession Recording Serverを信頼できず、接続できません。

Session Recording Server上のサーバー証明書がすべてのコンポーネントで信頼されていることを確かめてください。

- **名前の不一致。** Session Recording Serverをホストするコンピューターに割り当てられたサーバー証明書が、完全修飾ドメイン名を使用して作成されている場合、Session Recording Serverに接続するとき、接続するすべてのコンポーネントで完全修飾ドメイン名を使用する必要があります。サーバー証明書がNetBIOS名を使用して作成されている場合、Session Recording Serverに接続するとき、接続するすべてのコンポーネントでNetBIOS名を使用するように設定します。
- **失効した証明書。** サーバー証明書が失効している場合、HTTPSを介したSession Recording Serverへの接続は失敗します。Session Recording Serverをホストするコンピューターに割り当てられているサーバー証明書が有効で、失効していないことを確かめてください。録画したセッションのデジタル署名に同じ証明書を使用している場合は、Session Recording Serverをホストするコンピューターのイベントログに、証明書が失効したことを示すエラーメッセージまたは失効日が近いことを示す警告メッセージが記録されます。

# Session Recording Playerに障害が発生した場合の録画の検索

Jan 04, 2017

Session Recording Playerで録画を検索できないときは、次のエラーメッセージが表示される可能性があります。

- 「セッションの録画ファイルを検索できませんでした。リモート名を解決できませんでした。<servername>ここで、<servername>はSession Recording Playerで接続を試行しているサーバーの名前です。Session Recording PlayerはSession Recording Serverと通信することができません。この原因として、誤ったサーバー名が入力されているか、DNSでサーバー名を解決できていないという、2つの理由が考えられます。  
解決策： [ツール] メニューの [オプション] を選択します。 [接続] タブの [Session Recording Servers] ボックスの一覧のサーバー名が正しいことを確認します。サーバー名が正しい場合は、コマンドプロンプトでpingコマンドを実行し、名前を解決できるかどうかを確認します。Session Recording Serverが停止しているかオフラインのときにセッションの録画ファイルを検索すると、「リモートサーバーに接続できません」というエラーメッセージが返されます。
- 「リモートサーバーに接続できません。」このエラーは、Session Recording Serverが停止しているかオフラインのときに発生します。  
解決策： Session Recording Serverが接続していることを確かめます。
- 「Access denied error.」アクセス拒否のエラーは、ユーザーにセッションの録画ファイルを検索およびダウンロードする権限がない場合に発生します。  
解決策： Session Recording承認コンソールで、ユーザーをプレーヤーの役割に割り当てます。
- 「セッションの録画ファイルを検索できませんでした。「基になる接続が閉じられました。SSL/TLSの安全なチャンネルを確立できませんでした。」この例外は、Session Recording Serverで使用している証明書を署名したCA (Certificate Authority : 証明機関) がクライアントデバイスに信頼されていないか、クライアントデバイスにCA証明書がインストールされていないために発生します。  
解決策： Session Recording Playerがインストールされているワークステーションに、正しい、つまり信頼されているCA証明書をインストールします。
- 「リモートサーバーがエラーを返しました。HTTP 403 (アクセス不可)。」このエラーは、HTTP (セキュリティで保護されていないプロトコル) を使用して接続しようとしたときに発生する、標準のHTTPSエラーです。デフォルトでは、セキュリティで保護されている接続のみを受け入れるように設定されるため、サーバーにより接続が拒否されます。  
解決策： メニューバーで、 [ツール]、 [オプション]、 [接続] の順に選択します。 [Session Recordings Servers] ボックスの一覧でサーバーを選択し、 [変更] をクリックします。プロトコルを [HTTP] から [HTTPS] に変更します。

セッションの録画を知らせる通知メッセージがエンドユーザーに表示されているのに、Session Recording Playerで検索しても録画が見つからない場合は、MSMQに問題がある可能性があります。キューがSession Recording Server (ストレージマネージャー) に接続されていることを確認し、Webブラウザを使用して接続エラーが発生しないか確認します (MSMQの接続プロトコルとしてHTTPまたはHTTPSを使用している場合)。

キューが接続されていることを確認するには

1. Session Recording Agentをホストするサーバーにログインして、発信キューを表示します。
2. Session Recording Serverをホストするコンピューターへのキューが接続された状態であることを確認します。

- 接続を待っている状態で、多くのメッセージがキューにあり、プロトコルがHTTPまたはHTTPSの場合は ([Session Recording Agentのプロパティ] ダイアログボックスの [接続] タブで選択されているプロトコルに対応します)、手順3を実行します。
  - 接続済みの状態で、メッセージがキューにない場合は、Session Recording Serverをホストするサーバーに問題がある可能性があります。手順3を省略し、手順4を実行します。
3. キューに多くのメッセージがある場合は、Webブラウザを起動して次のアドレスを入力します。
- HTTPSで接続する場合：https://<servername>/msmq/private\$/CitrixSmAudData (ここで、<servername>はSession Recording Serverをホストするコンピューターの名前です)
  - HTTPで接続する場合：http://<servername>/msmq/private\$/CitrixSmAudData (ここで、<servername>はSession Recording Serverをホストするコンピューターの名前です)
- 「サーバーはセキュリティで保護された接続のみを受け付けます」というようなエラーが返される場合は、[Session Recording Agentのプロパティ] ダイアログボックスでMSMQプロトコルをHTTPSに変更します。そうではなく、Webサイトのセキュリティ証明書に問題があるというエラーが返される場合は、SSL/TLSのセキュアチャネルのための信頼関係に問題がある可能性があります。その場合は、正しいCA証明書をインストールするか、信頼されているCAを使用します。
4. キューにメッセージがない場合は、Session Recording Serverをホストするコンピューターにログオンし、専用キューを表示します。citrixsmalldataを選択します。キューに多くのメッセージがある場合は ([メッセージ数] 列を確認します)、Session Recording StorageManagerサービスが開始されていることを確認します。開始されていない場合は、サービスを再起動します。

# 通信プロトコルの変更

Jan 04, 2017

セキュリティ上の理由から、HTTPを通信プロトコルに使用することはお勧めできません。デフォルトでは、Session RecordingはHTTPSを使用して通信するように設定されます。HTTPSではなくHTTPを使用する場合は、いくつかの設定を変更する必要があります。

1. Session Recording Serverをホストするコンピューターにログオンし、IISで、Session Recording Brokerの接続のセキュリティを保護しないようにします。
  2. 各サーバーの [Session Recording Agentのプロパティ] ダイアログボックスで、プロトコル設定をHTTPSからHTTPに変更します。
    1. Session Recording Agentがインストールされている各サーバーにログオンします。
    2. [スタート] ボタンをクリックし、[Session Recording Agentのプロパティ] を選択します。
    3. [Session Recording Agentのプロパティ] ダイアログボックスで [接続] タブをクリックします。
    4. [Session Recording Broker] で [プロトコル] ボックスの一覧から [HTTP] を選択し、[OK] をクリックして変更を受け入れます。サービスの再起動を促すメッセージが表示されたら、[はい] を選択します。
  3. Session Recording Playerの設定でプロトコルをHTTPSからHTTPに変更します。
    1. Session Recording Playerがインストールされている各ワークステーションにログオンします。
    2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
    3. [Session Recording Player] メニューバーで [ツール]、[オプション]、[接続] の順に選択し、サーバーを選択して [変更] をクリックします。
    4. [プロトコル] ボックスの一覧から [HTTP] を選択し、[OK] を2回クリックして、変更を受け入れてダイアログボックスを閉じます。
  4. Session Recording Policy Consoleの設定でプロトコルをHTTPSからHTTPに変更します。
    1. Session Recording Policy Consoleがインストールされているサーバーにログオンします。
    2. [スタート] ボタンをクリックし、[Session Recording Policy Console] を選択します。
    3. [プロトコル] ボックスの一覧から [HTTP] を選択し、[OK] をクリックして接続します。接続が確立するとこの設定が保存され、次にSession Recording Policy Consoleを起動するときにも使用されます。
- 
1. Session Recording Serverをホストするコンピューターにログオンし、IISで、Session Recording Brokerの接続のセキュリティを保護します。
  2. 各サーバーの [Session Recording Agentのプロパティ] ダイアログボックスで、プロトコル設定をHTTPからHTTPSに変更します。
    1. Session Recording Agentがインストールされている各サーバーにログオンします。
    2. [スタート] ボタンをクリックし、[Session Recording Agentのプロパティ] を選択します。
    3. [Session Recording Agentのプロパティ] ダイアログボックスで [接続] タブをクリックします。
    4. [Session Recording Broker] で [プロトコル] ボックスの一覧から [HTTPS] を選択し、[OK] をクリックして変更を受け入れます。サービスの再起動を促すメッセージが表示されたら、[はい] を選択します。
  3. Session Recording Playerの設定でプロトコルをHTTPからHTTPSに変更します。
    1. Session Recording Playerがインストールされている各ワークステーションにログオンします。
    2. [スタート] ボタンをクリックし、[Session Recording Player] を選択します。
    3. [Session Recording Player] メニューバーで [ツール]、[オプション]、[接続] の順に選択し、サーバーを選択して [変更] をクリックします。

4. [プロトコル] ボックスの一覧から [HTTPS] を選択し、[OK] を2回クリックして、変更を受け入れてダイアログボックスを閉じます。
4. Session Recording Policy Consoleの設定でプロトコルをHTTPからHTTPSに変更します。
  1. Session Recording Policy Consoleがインストールされているサーバーにログオンします。
  2. [スタート] ボタンをクリックし、[Session Recording Policy Console] を選択します。
  3. [プロトコル] ボックスの一覧から [HTTPS] を選択し、[OK] をクリックして接続します。接続が確立するとこの設定が保存され、次にSession Recording Policy Consoleを起動するときにも使用されます。

# リファレンス：データベースレコードの管理

Jan 04, 2017

ICAログデータベース (ICLDB) ユーティリティは、データベース用のコマンドラインユーティリティで、セッションの録画のデータベースレコードを操作するために使用します。このユーティリティは、Session Recording Serverソフトウェアをホストするサーバーの<drive>\Citrix\SessionRecording\Server\Binフォルダーに、Session Recordingと合わせてインストールされます。

次の表に、ICLDBユーティリティで使用できるコマンドとオプションの一覧を示します。コマンドは次の形式で入力します。

icldb[version|locate|dormant|import|archive|remove|removeall][/] [/f] [/s] [/?]

注：ユーティリティのオンラインヘルプにさらに詳しい説明が記載されています。オンラインヘルプにアクセスするには、コマンドプロンプトで<drive>\Citrix\SessionRecording\Server\Binフォルダーに移動し、icldb /?と入力します。特定のコマンドのオンラインヘルプにアクセスするには、icldb <command> /?と入力します。

| コマンド    | 説明                                                                                                                                                                                             |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| archive | 指定された保有期間を過ぎたセッションの録画ファイルをアーカイブします。<br>このコマンドを使用してファイルをアーカイブします。                                                                                                                               |
| dormant | 休止状態とみなされるセッションの録画ファイルの数またはファイル名を表示します。休止ファイルとは、データの損失のために不完全なセッションの録画ファイルです。<br>このコマンドを使用してデータの損失があるかどうかを検証します。休止状態のセッションの録画ファイルの検索対象として、データベース全体を指定することも、日、時間、または分単位で、録画が行われた期間を指定することもできます。 |
| import  | セッションの録画ファイルをSession Recordingデータベースにインポートします。<br>このコマンドを使用して、データベースレコードを失ったときにデータベースを再構築します。<br><br>また、このコマンドを使用して、データベースをマージします。2つのデータベースがある場合は、一方のデータベースからファイルをインポートできます。                   |
| locate  | ファイルIDを条件として、セッションの録画ファイルを検索しフルパスを表示します。<br>このコマンドを使用して、セッションの録画ファイルの格納場所を検索します。<br><br>このコマンドは、特定のファイルを条件にデータベースが最新の状態かどうかを検証する手段としても使用できます。                                                  |
| remove  | セッションの録画ファイルへの参照をデータベースから削除します。<br>このコマンドを使用して、データベースをクリーンアップします。ただし、注意して使用してください。条件として使用する保有期間を指定します。<br><br>関連付けられている物理ファイルを削除することもできます。                                                     |

|                        |                                                                                                                                                                                                                                      |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>removeall</code> | <p><b>説明</b> ションの録画ファイルへのすべての参照をSession Recordingデータベースから削除し、データベースを元の状態に戻します。実際の物理ファイルは削除されません。ただし、Session Recording Playerでファイルを検索することはできなくなります。このコマンドを使用して、データベースをクリーンアップします。ただし、注意して使用してください。削除された参照はバックアップから復元しない限り元に戻せません。</p> |
| <code>version</code>   | Session Recordingデータベースのスキーマバージョンを表示します。                                                                                                                                                                                             |
| <code>/l</code>        | 結果とエラーをWindowsのイベントログに記録します。                                                                                                                                                                                                         |
| <code>/f</code>        | プロンプトを表示せずにコマンドを強制的に実行します。                                                                                                                                                                                                           |
| <code>/s</code>        | 著作権のメッセージを非表示にします。                                                                                                                                                                                                                   |
| <code>/?</code>        | コマンドのオンラインヘルプを表示します。                                                                                                                                                                                                                 |



# サードパーティ製品についての通知

Jan 04, 2017

Session Recordingには、以下の条項に基づいてライセンスを許諾されるサードパーティ製のソフトウェアコンポーネントが含まれる可能性があります。この一覧は掲載の日付にサードパーティ製ソフトウェアを使用して生成されたものです。この一覧は製品の特定のバージョンで変更される可能性があり、不完全である可能性があります。また、現状のままで提供されます。適用法令により許容される限度において、Citrixおよびその製品供給者は、一覧、その正確性もしくは完全性、または一覧の使用もしくは配布から得られる結果に関して、明示的、黙示的、法定を問わず、いかなる表明も保証も行いません。一覧を使用または配布することにより、お客様は、Citrixがいかなる場合においても、一覧の使用または配布に起因する、特別、直接、間接、もしくは二次的損害賠償、またはその他の損害賠償について、一切の責任を負わないことに合意します。

MMC .NET Library

Common Public License, Version 1.0に基づいてライセンスを許諾されます。

# Personal vDisk

Jan 04, 2017

Personal vDiskのユーザーデータ領域およびアプリケーション領域に対するユーザーによる変更は、Personal vDiskの診断ツールを使って監視できます。これらの変更には、ユーザーがインストールしたアプリケーションや修正したファイルが含まれます。変更内容はいくつかのレポートに保存されます。

1. 監視するマシン上で、C:\Program Files\Citrix\personal vDisk\bin\CtxPvdDiag.exeを実行します。
2. レポートとログを保存する場所を参照し、生成するレポートを選択して[OK] をクリックします。次のレポートを使用できます。

| レポートまたはログ       | 生成されるファイル                                          | 監視される変更                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SOFTWAREハイブレポート | Software.Dat.Report.txt、<br>Software.Dat.delta.txt | Software.Dat.Report.txtには<br>HKEY_LOCAL_MACHINE\Softwareハイブに対するユーザーの変更が記録されます。以下のセクションがありません。 <ul style="list-style-type: none"> <li>● <b>List of Applications installed on the base</b> — レイヤー0にインストールされたアプリケーション。</li> <li>● <b>List of user installed software</b> — vDiskのアプリケーション領域にインストールされたアプリケーション。</li> <li>● <b>List of software uninstalled by user</b> — レイヤー0から削除されたアプリケーション。</li> </ul> Software.Dat.delta.txtに記録される情報については、この表の「ハイブデルタレポート」を参照してください。 |
| SYSTEMハイブレポート   | SYSTEM.CurrentControlSet.DAT.Report.txt            | このファイルにはHKEY_LOCAL_MACHINE\Systemハイブに対するユーザーの変更が記録されます。以下のセクションがあります。 <ul style="list-style-type: none"> <li>● <b>List of user installed services</b> — インストールされたサービスおよびドライバ。</li> <li>● <b>Startup of following services were changed</b> — 起動の種類が変更されたサービスおよびドライバ。</li> </ul>                                                                                                                                                                                      |
| SECURITYハイブレポート | SECURITY.DAT.Report.txt                            | このファイルには、HKEY_LOCAL_MACHINE\Securityハイブに対するユーザーの変更が記録されます。                                                                                                                                                                                                                                                                                                                                                                                                           |
| SAMハイブレポート      | SAM.DAT.Report.txt                                 | このファイルには、HKEY_LOCAL_MACHINE\SAMハイブに対するユーザーの変更が記録されます。                                                                                                                                                                                                                                                                                                                                                                                                                |
| ハイブデルタレポート      | Software.Dat.delta.txt                             | このファイルには、HKEY_LOCAL_MACHINE\Softwareハイブでユーザーが追加、削除または変更したすべてのレジストリキーや値が記録されます。                                                                                                                                                                                                                                                                                                                                                                                       |

|                                        |                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>レポートまたはログ<br/>Personal vDiskのログ</p> | <p>生成されるファイル<br/>Pvd-Ivmsupervisor.log、<br/>PvDActivation.log、PvDSvc.log、<br/>PvDWMl.log、SysVol-IvmSupervisor.log、<br/>vDeskService-&lt;#&gt;.log</p> | <p>監視される変更<br/>これらのファイルは、デフォルトでは<br/>P:\Users\Local\Temp\PVDLOGSに生成され<br/>ますが、選択した場所に移動されます。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <p>Windowsオペレーティングシステム (OS) のログ</p>    | <p>EvtLog_App.xml、EvtLog_System.xml、<br/>setupapi.app.log、setuperr.log、<br/>setupapi.dev.log、msinfo.txt</p>                                           | <p>EvtLog_App.xmlおよびEvtLog_System.xmlは、Personal vDiskボリュームから生成されるXML形式のアプリケーションおよびシステムイベントログです。</p> <p>Setupapi.app.logおよびsetuperr.logには、Personal vDiskのインストールでmsiexec.exeを実行したときからのログメッセージが記録されます。</p> <p>Setupapi.dev.logには、デバイスインストールログメッセージが記録されます。</p> <p>Msinfo.txtには、msinfo32.exeからの出力が記録されます。この出力について詳しくは、Microsoft社のドキュメントを参照してください。</p>                                                                                                                                                                                                                                                                                                                                                          |
| <p>ファイルシステムレポート</p>                    | <p>FileSystemReport.txt</p>                                                                                                                           | <p>このファイルには、ファイルシステムに対するユーザーの変更が記録されます。以下のセクションがあります。</p> <ul style="list-style-type: none"> <li>● <b>Files Relocated</b> — Personal vDiskに移動されたレイヤー0のファイル。レイヤー0のファイルとは、Personal vDiskが接続されたマシンによりマスターイメージから継承されたファイルを指します。</li> <li>● <b>Files Removed</b> — ユーザーの操作 (アプリケーションの削除など) によって非表示になったレイヤー0のファイル。</li> <li>● <b>Files Added (MOF、INF、SYS)</b> — Personal vDiskに追加された拡張子.mof、.inf、または.sysを持つファイル (Visual Studio 2010などのアプリケーションのインストールにより登録された自動回復用のMOFファイルなど)。</li> <li>● <b>Files Added Other</b> — Personal vDiskに追加されたその他のファイル (アプリケーションのインストールにより追加されたファイルなど)。</li> <li>● <b>Base Files Modified But Not Relocated</b> — 変更されたレイヤー0のファイルで、Personal vDiskカーネルモードドライバーにより移動されないもの。</li> </ul> |

# 構成ログ

Jan 04, 2017

構成ログ機能では、管理者によるサイト構成の変更やその他の管理操作がデータベースに記録されます。このログは、以下の目的で使用できます。

- 構成変更の履歴を確認して問題の診断およびトラブルシューティングを行う。
- 変更管理の補助および構成の追跡を行う。
- 管理アクティビティのレポートを生成する。

Citrix Studioでは、構成ログの基本設定を変更したり、構成ログを表示したり、HTMLおよびCSV形式のレポートを生成したりできます。日範囲および全文検索の結果により構成ログ表示をフィルターできます。必須ログ機能を有効にすると、ログが記録可能になるまで管理者による構成の変更が禁止されます。適切な権限を持つ管理者は、構成ログのエントリを削除できます。構成ログ機能では、ログの内容を編集することはできません。

構成ログでは、PowerShell 2.0 SDKとConfiguration Logging Serviceが使用されます。Configuration Logging Serviceは、サイト内のすべてのController上で実行されます。任意のControllerに障害が発生しても、ほかのControllerが自動的にログ要求を処理します。

デフォルトでは、構成ログ機能は有効で、サイト作成時に作成されたデータベース（サイト構成データベース）が使用されます。ただし、サイトを作成した後で、構成ログで使用するデータベースの場所をできるだけ早く変更することを強く推奨します。構成ログデータベースでは、サイト構成データベースと同じ高可用性機能がサポートされます。

構成ログにアクセスするには、[ログ基本設定を編集] および [構成ログを表示] 権限が必要です。

構成ログの言語には、作成時のロケールが適用されます。たとえば、英語で作成されたログは、管理者側のロケールには関係なく英語で表示されます。

構成ログには、Studio、Director、およびPowerShellスクリプトから開始された構成の変更および管理アクティビティのログが記録されます。以下の項目に対する作成、編集、削除などの操作が構成ログに記録されます。

- マシンカタログ
- デリバリーグループ（電源管理設定の変更を含む）
- 管理者の役割とスコープ
- ホストのリソースおよび接続
- Studioで構成するCitrixポリシー

ログが記録される管理変更の例には次のものがあります。

- 仮想マシンまたはユーザーのデスクトップの電源管理
- StudioまたはDirectorからユーザーへのメッセージ送信

次の操作はログに記録されません。

- 仮想マシンのプール管理電源オンなどの自動操作。
- グループポリシー管理コンソール（GPMC）でのポリシー操作。これらの操作のログはMicrosoftのツールを使って表示できます。
- レジストリによる変更、データベースの直接的な変更、およびStudio、Director、PowerShell以外での変更。
- 展開の初期化後、最初のConfiguration Logging ServiceインスタンスがConfiguration Serviceに登録されたときに構成ログが有効になります。このため、構成の初期のアクティビティが記録されない場合があります（ハイパーバイザーの初期化時にデータベーススキーマが取得および適用される場合など）。

# 構成ログの管理

Jan 04, 2017

デフォルトでは、サイトの作成時に作成されたデータベース（サイト構成データベース）に構成ログが記録されます。ただし、サイトを作成した後で構成ログ用のデータベース（およびデフォルトでサイト構成データベースを使用するMonitoring Service用のデータベース）の場所を変更することをお勧めします。これは、以下の理由によります。

- 構成ログデータベースのバックアップ方針が、サイト構成データベースのバックアップ方針と異なる場合があります。
- 構成ログ（およびMonitoring Service）で収集されるデータの量によっては、サイト構成データベース用の領域が不足する場合があります。
- データベースを分散させると、単一ポイント障害の問題が解消されます。

注：構成ログをサポートしない製品エディションでは、Studioに [ログ] ノードが表示されません。詳しくは、「[XenDesktop 7.6 and XenApp 7.6 Features and Entitlements](#)」を参照してください。

## 構成ログおよび必須ログの有効化および無効化

構成ログ機能はデフォルトで有効になっており、必須ログ機能は無効になっています。

1. Studioのナビゲーションペインで [ログ] ノードを選択します。
2. [操作] ペインの [基本設定] を選択します。 [ログ設定] ダイアログボックスが開き、データベースに関する情報と、構成ログおよび必須ログ機能の有効/無効が表示されます。
  - 構成ログを有効にするには、 [有効] をクリックします。これがデフォルトの設定です。データベースに書き込みができない場合、ログ情報は破棄されますが構成内容は正しく反映されます。
  - 構成ログを無効にするには、 [無効] をクリックします。それまでに記録されたログの内容は、PowerShell SDKで読み取ることができます。
  - 必須ログ機能を有効にするには、 [データベースが切断されていても構成変更を許可する] チェックボックスをオフにします。構成ログデータベースへの書き込みが可能になるまで、管理者は管理タスクを実行できなくなります（構成ログが優先されます）。  
必須ログ機能は、構成ログが有効な場合、つまり [有効] が選択されている場合にのみ有効にできます。Configuration Logging Serviceに障害が発生して、しかも高可用性が無効な場合、必須ログが有効になります。このような場合、構成ログデータベースに記録されるようなタスクは実行できなくなります。
  - 必須ログ機能を無効にするには、 [データベースが切断されていても構成変更を許可する] チェックボックスをオンにします。構成ログデータベースにアクセスできない場合でも、管理者は構成の変更やその他の管理タスクを実行できます（管理タスクが優先されます）。これがデフォルトの設定です。

## 構成ログデータベースの場所の変更

注：必須ログ機能が有効になっている場合、データベースの場所を変更することはできません。データベースの変更時に短時間データベースから切断されるためです。

1. サポートされるバージョンのSQL Serverを使用してデータベースサーバーを作成します。
2. Studioのナビゲーションペインで [ログ] ノードを選択します。
3. [操作] ペインの [基本設定] を選択します。
4. [ログ設定] ダイアログボックスで [ログデータベースの変更] をクリックします。
5. [ログデータベースの変更] ダイアログボックスで、新しいデータベースサーバーの場所とデータベース名を指定します。ここでは、移動先データベースの種類と構成に応じて、以下の内容を入力します。

| データベースの種類 | 入力内容       | データベースの構成                                |
|-----------|------------|------------------------------------------|
| スタンドアロンまた | servername | デフォルトのインスタンスが使用され、SQL Serverでデフォルトのポートが使 |

| データベースの種類 | 入力内容          | 構成                                                        |
|-----------|---------------|-----------------------------------------------------------|
|           | <servername>\ | 名前付きのインスタンスが使用され、SQL Serverでデフォルトのポートが使用されている場合。          |
|           | ,             | デフォルトのインスタンスが使用され、SQL Serverでカスタムポートが使用されている場合（コンマが必要です）。 |
| その他       |               | データベースがクラスター化されている場合。                                     |
|           |               | Always-Onデータベースが構成されている場合。                                |

- Studioで自動的にデータベースを作成する場合は、[OK] をクリックします。確認のメッセージが表示され、[OK] をクリックするとデータベースが自動的に作成されます。現在のStudioユーザーの資格情報を使ってデータベースへのアクセスが試行されます。アクセスに失敗すると、データベースにアクセスするための資格情報を入力する画面が開きます。アクセスに成功すると、データベーススキーマがデータベースにアップロードされます（資格情報はデータベース作成時のみ保持されます）。
- データベースを手動で作成する場合は、[データベーススクリプトの生成] をクリックします。生成されるスクリプトにはデータベースを手動で作成するためのコマンドが記述されます。スキーマをアップロードする前に、データベースが空であること、および1人以上のユーザーがそのデータベースにアクセスでき、変更できることを確認してください。

変更前のデータベース内の構成ログデータは変更後のデータベースにインポートされません。構成ログデータベースの場所を変更する場合、変更前のデータベースの内容は集約されなくなります。変更後の構成ログデータベースの最初にデータベースの変更を示すログが記録されますが、変更前のデータベースの場所は記録されません。

### 構成ログの内容の表示

管理者が構成の変更などの管理アクティビティを開始すると、StudioやDirectorによって作成された高レベル操作がStudioの中央ペインの上部に表示されます。高レベル操作により1つまたは複数のサービスおよびSDKの呼び出しが実行されます。これは、低レベル操作です。中央ペインの上部で高レベル操作を選択すると、中央ペインの下部に低レベル操作が表示されません。

操作が完了する前に失敗すると、たとえば開始レコードに対応する停止レコードがないなど、データベースでログ操作が完了しない場合があります。このような場合、情報不足であることがログに示されます。時間の範囲を指定してログを表示する場合、未完了のログが表示される場合があります。たとえば、直近5日間のログを表示するときにその5日間に開始時間のみが含まれ、終了時間が含まれていない場合も、その操作のログが表示されます。

PowerShellコマンドレットを呼び出すスクリプトを使う場合、親の高レベル操作を指定せずに低レベル操作を作成すると、構成ログにより代わりの高レベルの操作が作成されます。

構成ログの内容を表示するには、Studioのナビゲーションペインで[ログ] ノードを選択します。デフォルトでは、中央ペインにログコンテンツが時系列順に（最新のエントリが最初に）表示されます。

| 表示条件      | 表示条件の指定方法                                                                                    |
|-----------|----------------------------------------------------------------------------------------------|
| 特定の文字列で検索 | 中央ペイン上部にある [検索] ボックスに文字列を入力します。入力した文字列を含んでいるエントリのみが表示されます。通常のログ表示に戻すには、[検索] ボックスの文字列をクリアします。 |
| 列見出し      | 列見出しをクリックして、表示をその列のデータで並べ替えます。                                                               |

|             |                                                            |
|-------------|------------------------------------------------------------|
| 表示条件<br>日範囲 | 表示条件の指定方法<br>中央ペインの上部で [検索] ボックスの横にあるドロップダウンの一覧から期間を選択します。 |
|-------------|------------------------------------------------------------|

## レポートの生成

構成ログデータをCSVおよびHTML形式のレポートとして書き出すことができます。

- CSV形式のレポートには、指定した期間のすべてのログデータが書き込まれます。データベースの階層データが単一のCSVテーブルとして出力されます。データの特定の要素に基づいて並べ替えられたものではありません。書式も適用されず、読み取りやすさについても考慮されていません。レポートファイル (MyReport) には、汎用的な書式でデータが書き出されます。CSVファイルはデータのアーカイブ化や、レポート機能またはMicrosoft Excelなどデータ操作ツールのデータソースとして使用されます。
- HTML形式のレポートには、指定した期間のログデータが判読可能な形式で書き込まれます。変更内容の確認が容易な、構造的でナビゲーション可能なレポートです。HTMLレポートでは、概要 (Summary) および詳細 (Details) の2つのファイルが生成されます。概要レポートには、各操作の実行日時、操作主、および操作結果など、より高レベルな情報が一覧で表示されます。各操作項目の横にある [詳細] リンクをクリックすると詳細ファイルが開き、より低いレベルの操作に関する情報を参照できます。

構成ログレポートを生成するには、Studioのナビゲーションペインで[ログ] ノードを選択し、[操作] ペインの[カスタムレポートの作成] を選択します。

- レポートの日付の範囲を選択します。
- レポート形式として、[CSVファイル]、[HTML]、または[両方]を選択します。
- レポートを保存する場所を参照します。

## 構成ログの内容の削除

構成ログを削除するには、特定の委任管理権限およびSQL Serverデータベース権限が必要です。

- **委任管理権限** — 展開の構成を読み取るための権限が必要です。組み込みのすべての管理権限を実行できる管理者には、この権限があります。カスタムの役割では、[そのほかの権限] カテゴリで [読み取り専用] または [管理] 権限を選択する必要があります。

構成ログデータを削除する前にバックアップを作成するには、[ログ] カテゴリで [読み取り専用] または [管理] 権限を選択する必要もあります。

- **SQL Serverデータベース** — データベースから記録を削除するための権限を持つアカウントが必要です。次のいずれかの方法を使用します。
  - データベースに対するすべての権限を持つsysadminサーバーロールを持つSQL Serverデータベースログインを使用します。また、serveradminまたはsetupadminサーバーロールでも削除操作を実行できます。
  - 高度なセキュリティが必要な環境では、データベースからレコードを削除する権限を持つデータベースユーザーにマップされた非sysadminデータベースログインを使用します。
    1. SQL Server Management Studioで、sysadmin以外のサーバーロールを持つSQL Serverログインを作成します。
    2. 作成したログインをデータベースのユーザーにマップします。SQL Serverにより、ログインと同じ名前のユーザーがデータベースに作成されます。
    3. データベースロールのメンバーシップとして、このデータベースユーザーにConfigurationLoggingSchema\_ROLEまたはdb\_ownerのロールを指定します。
詳しくは、SQL Server Management Studioのドキュメントを参照してください。

構成ログを削除するには、以下の手順に従います。

1. Studioのナビゲーションペインで [ログ] ノードを選択します。
2. [操作] ペインの [ログの削除] を選択します。
3. 削除する前にログのバックアップを作成するかどうかを確認するメッセージが表示されます。バックアップを作成する場

合は、バックアップを保存する場所を参照します。バックアップはCSVファイルとして作成されます。

構成ログを削除すると、その削除操作が最初のエントリとしてログに記録されます。このエントリには、いつだれがログを削除したのかが記述されます。



# Monitor Service OData API

Jan 04, 2017

管理者は、Citrix Directorのコンソールで表示した履歴データを、Monitor ServiceのAPIを使って照会できます。このAPIを使用すると、次の操作を行えます。

- 将来の計画のために履歴傾向を分析する。
- 接続やマシンの障害に対する詳細なトラブルシューティングを実行する。
- ほかのツールおよびプロセスに情報をインポートする（Microsoft ExcelのPowerPivotテーブルを使って別の方法でデータを表示するなど）。
- APIで提供されるデータ上にカスタムユーザーインターフェイスを構築する。

Monitor Service APIではOpen Data (OData) プロトコルが使用されます。これはデータを照会および更新するためのWebプロトコルで、HTTPなどのWebテクノロジーに基づいています。ODataプロトコルについては、<http://www.odata.org>を参照してください。

Monitor Service APIは、処理および統合時に読み込まれるWindows Communication Foundation (WCF) Data Servicesを使用して、SQL Serverデータベース上に構築されます。WCFを使用して2つのエンドポイントがwsHttpBindingで公開されます。ベースアドレスは「<http://{dc-host}/Citrix/Monitor/OData/v2>」です。また、SSLを使ってエンドポイントを保護することもできます。詳しくは、「[SSLを使ったエンドポイントのセキュリティ保護](#)」を参照してください。

1. Dataエンドポイントは、データベースエンティティへの直接読み取り専用アクセスを公開し、ODataクエリ言語を使用してアクセスすることができます。このエンドポイントは、フィルタリングおよび列の選択により高度に柔軟なアクセスを行えます。Data APIのURIは「<http://{dc-host}/Citrix/Monitor/OData/v2/Data>」です。Monitor Serviceデータへのアクセスについては詳しくは、「[APIを使ったデータアクセス](#)」を参照してください。
2. Methodsエンドポイントは、[ダッシュボード] ページや [傾向] ページでの照会など、Citrix Directorで複雑なグループ化や高性能な標準を必要とするデータを取得するためのサービス操作を公開します。Methods APIのURIは「<http://{dc-host}/Citrix/Monitor/OData/v2/Methods>」です。MethodsはDirectorでのみ使用され、通常Citrix製品の管理者やユーザーが使用することはありません。したがって、ここでは説明しません。

## このリリースでの新機能

XenApp/XenDesktop 7.6に付属のAPIにより、以下の新機能が提供されます。

- **Hotfixインベントリ**。Directorの [ユーザーの詳細] ビューまたは [マシン] ビューを使用すると、マシンにインストールされているCitrix Hotfixの一覧を表示できます。このAPIを使用すると、このデータを抽出してカスタムレポート（サイト全体でインストールされているHotfixの状態など）を作成したり、分析エンジンに取り込んだりできます。いくつかのクラスが追加され、MachineクラスにControllerやVDAにインストールされたCitrix Hotfixの追跡機能が追加されました。
- **匿名セッションのトラブルシューティング**。プールされたローカルユーザーアカウントのセットを使用してセッションを実行できます。このAPIでは、Sessionクラスに新しいIsAnonymousプロパティ（デフォルト値：FALSE）が追加されています。
- **ホストされるアプリケーションの使用量レポート**。Directorでは、ホストされるアプリケーションの経時的な使用状況を示すキャパシティレポートが提供されます。このAPIを使用すると、ユーザーセッションで実行されている各アプリケーションインスタンスの詳細をレポートできます。

このAPIの新機能については、<http://support.citrix.com/help/monitorserviceapi/7.6/>のAPIリファレンスを参照してください。

GetSessionSummaryメソッドは、このリリースでは廃止されました。

# APIを使ったデータアクセス

Jan 04, 2017

以下の種類のデータは、Monitor Service APIを介して使用できます。

- 接続エラーに関するデータ
- エラー状態のマシン
- セッションの使用
- ログオン処理時間
- 負荷分散データ
- マシンに適用されたHotfix
- ホストされたアプリケーションの使用

データオブジェクトの説明について詳しくは、<http://blogs.citrix.com/2013/08/27/xendesktop-7-monitor-service-what-data-is-available/>を参照してください。

Monitor Service OData APIを使用するには、XenAppまたはXenDesktopの管理者である必要があります。APIを呼び出すには、読み取り専用の特権が必要です。ただし、返されるデータは、その管理者の役割と権限によって異なる場合があります。たとえば、デリバリーグループ管理者はMonitor Service APIを呼び出すことができますが、取得可能なデータはCitrix Studioでセットアップされるデリバリーグループアクセスにより制御されます。XenApp/XenDesktop管理者の役割と権限について詳しくは、「[管理権限の委任](#)」を参照してください。

## データのクエリ

Monitor Service APIは、ODataコンシューマーでアクセスできるRESTベースのAPIです。ODataコンシューマーは、ODataプロトコルで公開されたデータを消費するアプリケーションを指します。シンプルなWebブラウザからODataプロトコルのすべての機能を実行できるカスタムアプリケーションまでさまざまなODataコンシューマーがあります。ODataコンシューマーについて詳しくは、<http://www.odata.org/ecosystem#consumers>を参照してください（英文）。

Monitor Serviceデータモデルのすべての部分はアクセス可能で、URL上でフィルタリングできます。ODataでは、URL形式のクエリ言語を使用してサービスからエントリを抽出できます。詳しくは、<http://msdn.microsoft.com/ja-jp/library/ff478141.aspx>を参照してください。

クエリはサーバー側で処理され、クライアント側でODataプロトコルを使ってさらに詳細にフィルタリングできます。

取得されるデータは、集計データ（サマリーテーブル）、オブジェクト（マシン、セッションなど）の現在の状態、およびログデータ（接続イベントなどの履歴）の3つのカテゴリに分類されます。

注：ODataプロトコルでは列挙（enum）はサポートされず、整数（integer）が使用されます。Monitor Service OData APIによって返される値については、<http://support.citrix.com/help/monitorserviceapi/7.6/>を参照してください。

## データ値の集計

Monitor Serviceは、ユーザーセッション使用状況、ユーザーログオンの処理性能の詳細、セッションの負荷分散の詳細、および接続とマシンのエラー情報を含む、さまざまなデータを収集します。データはカテゴリにより異なる方法で集計されます。OData Method APIを使って示されたデータ値の集計を理解することは、データの解釈に不可欠です。次に例を示します。

- 接続セッション（Connected Session）やマシン障害（Machine Failure）は一定の期間の状態を示すため、その期間内の最大値として公開されます。
- ログオン処理時間（LogOn Duration）は時間の長さを示す指標であるため、期間内の平均として公開されます。

- ログオン数 (LogOn Count) および接続障害 (Connection Failure) は一定の期間に発生した数を示し、期間内の合計値として公開されます

## 同時接続データの評価

重複しているセッションは同時発生していると考えする必要があります。ただし、間隔として1分を指定した場合、1分以内に発生するすべてのセッションは（重複しているかしていないかに関係なく）すべて同時であるとみなされます。つまり、間隔のサイズが非常に小さいため、精度の計算に関係するパフォーマンス上のオーバーヘッドを考慮する必要はありません。2つのセッションがその1時間内の別々の1分間に発生する場合、それらは重複しているとはみなされません。

## サマリーテーブルと生データの相関関係

データモデルでは、以下の2つの方法でメトリックスが示されます。

- サマリーテーブルでは、分単位、時間単位、および日単位のメトリックスを集計したものが示されます。
- 生データは、セッション、接続、アプリケーション、およびその他のほかのオブジェクト内で記録された個々のイベントまたは現在の状態を示します。

データをAPIコール間またはそのデータモデル内で関連付けるときは、以下の概念および制限事項を考慮してください。

- **未完の間隔にはサマリーデータがありません。**メトリックスサマリーは長時間での履歴傾向を示すためのものであり、完了した間隔のサマリーテーブルに集計されます。データ収集の開始時や終了時のサマリーデータはありません。1日（間隔=1440）の集計値の場合、最初と最後の未完の1日にはデータがないことを意味します。これらの未完の間隔に生データが存在しても、そのデータが集計されることはありません。各データ粒度の最初と最後の集計間隔は、各サマリーテーブルから最小と最大のSummaryDateを取得することで決定できます。SummaryDate列は、間隔の開始時を示します。Granularity列はその集計データの間隔の長さを示します。
- **時間による関連付け。**前述のように、メトリックスは完了した間隔のサマリーテーブルに集計されます。これらの値は履歴傾向を知る目的で使用できますが、生イベントの方が集計された値よりも傾向分析に適切な状態を示している場合があります。集計値と生データとを時間ベースで比較する場合、未完の間隔や間隔の最初と最後にサマリーデータがないことを考慮する必要があります。
- **欠落イベントまたは潜在イベント集計期間で欠落または潜在しているイベントがあると、サマリーテーブルに集計されたメトリックスが正確でない場合があります。**Monitor Serviceでは現在の状態の正確な維持が試行されますが、過去にさかのぼって欠落イベントや潜在イベントをサマリーテーブルに再集計することはありません。
- **接続の高可用性。**接続の高可用性により、現在の接続のサマリーデータ数に差異が生じることがありますが、セッションインスタンスは生データ内で実行されています。
- **データの保持期間。**サマリーテーブルのデータは、生イベントデータとは異なるクリーンアップ（グルーミング）スケジュールで保持されます。このため、サマリーテーブルまたは生テーブルのクリーンアップにより、データが消去されている場合があります。データの保持期間は、サマリーデータの粒度によっても異なる場合があります。低い粒度（分単位）のデータは、高い粒度（日単位）のデータよりも早くクリーンアップされます。特定の粒度のデータが消去されていても、より高い粒度のデータが存在している場合があります。APIコールでは指定した粒度のデータのみが返されるため、データを取得できない場合でもその期間内のより高い粒度では取得できることがあります。
- **タイムゾーン。**格納されるメトリックスのタイムスタンプではUTCが使用されます。サマリーテーブルは1時間区切りのタイムゾーンごとに集計されます。1時間区切りのタイムゾーンに属さない場合は、データの集計先に不整合が生じることがあります。

## データの粒度と保持

Directorで取得される集計データの粒度は、要求された時間（T）の関数です。以下の規則があります。

- $0 < T \leq 1$  時間の場合は分単位の粒度
- $0 < T \leq 30$  日の場合は時間単位の粒度
- $T > 31$  日の場合は日単位の粒度

集計データから取得されないデータを要求すると、生のセッション (Session) および接続 (Connection) 情報から取得されます。このデータの量はすぐに大きくなるため、専用のスケジュールでクリーンアップされます。クリーンアップにより、意味のあるデータのみが長期間保持されます。これにより、レポートに必要な粒度を維持しながら良好なパフォーマンスが提供されます。Platinum以外のエディションでは、8日目からクリーンアップが開始されます。Platinum Editionでは、クリーンアップが開始されるまでの日数をカスタマイズできます。

クリーンアップは以下の設定により制御されます。

| 設定名                         | 対象データ                                                                             | デフォルト値 (日数)                       | アクセス方法                                      |
|-----------------------------|-----------------------------------------------------------------------------------|-----------------------------------|---------------------------------------------|
| GroomSessionsRetentionDays  | SessionレコードおよびSessionDetailレコード                                                   | 非Platinumユーザーは7日、Platinumユーザーは90日 | コマンドレット (set/get-monitorconfiguration)      |
| GroomSummariesRetentionDays | DesktopGroupSummaryレコード、FailureLogSummaryレコード、およびLoadIndexSummaryレコード。集計データ (日単位) | 非Platinumユーザーは7日、Platinumユーザーは90日 | コマンドレット (set/get-monitorconfiguration)      |
| GroomHourlyRetentionDays    | 集計データ (時間単位)                                                                      | 32日                               | Monitor.Configuration Database Table。下記注を参照 |
| GroomMinuteRetentionDays    | 集計データ (分単位)                                                                       | 3日                                | Monitor.Configuration Database Table。下記注を参照 |
| GroomFailuresRetentionDays  | MachineFailureLogレコードおよびConnectionFailureLogレコード                                  | 非Platinumユーザーは7日、Platinumユーザーは90日 | コマンドレット (set/get-monitorconfiguration)      |

| 設定名                                    | 対象データ                                                                                                                                                                          | デフォルト値                                  | アクセス方法                                 |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|----------------------------------------|
| GroomLoadIndexesRetentionDays          | LoadIndexレコード                                                                                                                                                                  | 非 Platinum ユーザーは7日、Platinum ユーザーは90日    | (set/get-monitorconfiguration)         |
| GroomDeletedRetentionDays              | LifecycleStateが「Deleted」であるMachineエンティティ、Catalogエンティティ、DesktopGroupエンティティ、およびHypervisorエンティティ。関連するSessionレコード、SessionDetailレコード、Summaryレコード、Failureレコード、またはLoadIndexレコードも削除される | 非 Platinum ユーザーは7日、Platinum ユーザーは90日    | コマンドレット (set/get-monitorconfiguration) |
| GroomMachineHotfixHistoryRetentionDays | VDAマシンおよびControllerマシンに適用されたHotfix                                                                                                                                             | 非 Platinum ユーザーおよび Platinum ユーザーの両方で90日 | コマンドレット (set/get-monitorconfiguration) |

注意：Monitor Serviceデータベースの値を変更した後でその値を適用するには、このサービスを再起動する必要があります。Monitor Serviceデータベースの値の変更は、Citrixサポート担当者からの指示があった場合のみ行ってください。

データを長期間保持すると、テーブルのサイズについて以下の影響が発生することがあります。

- **時間単位のデータ。**時間単位のデータを2年などの長期間保持すると、1000個のデリバリーグループがあるサイトではデータベースが以下の数式に基づいて増大します。  
「1000個のデリバリーグループ×24時間/日×365日/年×2年 = 17,520,000行のデータ」集計テーブルのデータ量が多いため、パフォーマンスに大きな影響を及ぼします。ダッシュボードのデータがこのテーブルから取得されることを考慮すると、データベースサーバーに対する要求は高くなります。データ量が過度に多いと、パフォーマンスが大きく低下します。
- **セッションとイベントのデータ。**各セッションの開始時および接続/再接続時に収集されるデータです。大規模サイト（100,000ユーザーなど）では、このデータの量が急速に増加します。たとえば、これらのテーブルでは2年間で1TB以上データが保持され、高性能なエンタープライズレベルのデータベースが必要になります。

# SSLを使ったエンドポイントのセキュリティ保護

Jan 04, 2017

ここでは、Monitor Service OData APIエンドポイントを保護するためのSSLの使用方法について説明します。SSLを使用する場合、サイトのすべてのDelivery Controller上でSSLを構成する必要があります。SSLを構成したControllerと構成していないControllerを混在させることはできません。

Monitor ServiceエンドポイントをSSLで保護するには、以下の構成を行います。このための手順には、各サイトで1回のみ必要なものと、サイト内でMonitor Serviceをホストするすべてのマシン上で行うものがあります。次の手順に従います。

## パート1：システムへの証明書の登録

- 信頼された証明書マネージャーを使用して証明書を作成します。証明書は、OData SSLで使用するマシン上のポートに関連付ける必要があります。
- Monitor ServiceのSSL通信用のポートとしてこのポートを構成します。このための手順は、使用している環境とこれが証明書でどのように機能するかにより異なります。次の例では、ポート449に構成しています。
  - 証明書をポートに関連付けます。

```
netsh http add sslcert ipport=0.0.0.0:449 certhash=97bb629e50d556c80528f4991721ad4f28fb74e9 appid='{00000000-0000-0000-0000-000000000000}'
```

ヒント：PowerShellのコマンドウィンドウでは、上記のようにappIdのGUIDを一重引用符で囲む必要があります。一重引用符で囲まないと、コマンドが正しく動作しません。この例では、コマンドを読みやすくするために改行が追加されています。

## パート2：Monitor Service構成設定の変更

- サイト内の任意のDelivery Controller上で、次のPowerShellコマンドを1回実行します。これにより、Monitor ServiceがConfiguration Serviceから登録解除されます。

```
asnp citrix.* $serviceGroup = get-configregisteredinstance -servicetype Monitor | Select -First 1 ServiceGroupId remove-configservicegroup -ServiceGroupId $serviceGroup.ServiceGroupId
```
- サイト内のすべてのController上で、以下の操作を行います。
  - cmdプロンプトを開き、Citrix Monitorフォルダー（通常C:\Program Files\Citrix\Monitor\Service）に移動します。次のコマンドを実行します。

```
Citrix.Monitor.Exe -CONFIGUREFIREWALL -ODataPort 449 -RequireODataSsl
```
  - 次のPowerShellコマンドを実行します。

```
asnp citrix.* (if not already run within this window) get-MonitorServiceInstance | register-ConfigServiceInstance Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-MonitorServiceGroup
```

# 例

Jan 04, 2017

ここでは、OData APIを使用してMonitor Serviceデータをエクスポートする方法について例を挙げて説明します。

## 例1 - Raw XML

1. XenApp/XenDesktopサイトに対する適切な管理権限で実行しているWebブラウザで、各データセットのURLを開きます。Advanced Rest ClientアプリをインストールしたChromeブラウザの使用を推奨します。
2. ソースを表示します。

## 例2 - ExcelとPowerPivot

以下の手順は、Microsoft ExcelおよびPowerPivotがインストール済みであることを前提としています。

Excelを開きます（XenApp/XenDesktopサイトの適切な管理権限で実行します）。

Excel 2010では以下の操作を行います。

1. [PowerPivot] タブをクリックします。
2. [PowerPivotウィンドウ] をクリックします。
3. リボンの [データフィードから] をクリックします。
4. 接続の表示名としてわかりやすい名前（XenDesktop Monitoring Dataなど）を選択し、データフィードURLとして「<http://{{dc-host}}/Citrix/Monitor/OData/v2/Data>」（SSLを使用している場合は「[https:~](https://{{dc-host}}/Citrix/Monitor/OData/v2/Data)」）を入力します。
5. [次へ] をクリックします。
6. Excelにインポートするテーブルを選択し、[完了] をクリックします。データが取得されます。
7. これでPowerPivotのピボットテーブルやピボットグラフを使ってデータを表示および分析できるようになりました。詳しくは、Learning Center (<http://www.microsoft.com/en-us/bi/LearningCenter.aspx>) を参照してください。

Excel 2013では以下の操作を行います。

1. [データ] タブをクリックします。
2. [その他のソース]、[ODataデータフィード] の順に選択します。
3. データフィードURLとして「<http://{{dc-host}}/Citrix/Monitor/OData/v1/Data>」（SSLを使用している場合は「[https:~](https://{{dc-host}}/Citrix/Monitor/OData/v1/Data)」）を入力し、[次へ] をクリックします。
4. Excelにインポートするテーブルを選択し、[次へ] をクリックします。
5. デフォルトの名前を使用するか変更して、[完了] をクリックします。
6. [接続のみ] または [ピボットレポート] を選択します。データが取得されます。
7. これでPowerPivotのピボットテーブルやピボットグラフを使ってデータを表示および分析できるようになりました。詳しくは、Learning Center (<http://www.microsoft.com/en-us/bi/LearningCenter.aspx>) を参照してください。

## 例3 - LINQPad

以下の手順は、LINQPadがインストール済みであることを前提としています。

1. XenApp/XenDesktopサイトの適切な管理権限でLinqPadを実行します。  
ヒント：Delivery Controllerの上でダウンロード、インストール、および実行すると簡単です。
2. Add接続リンクをクリックします。
3. WCF Data Services 5.1 (OData 3) を選択し、[Next] をクリックします。
4. データフィードURLとして「<http://{{dc-host}}/Citrix/Monitor/OData/v2/Data>」（SSLを使用している場合は「[https:~](https://{{dc-host}}/Citrix/Monitor/OData/v2/Data)」）

～」)を入力します。必要な場合は、Delivery Controllerにアクセスするためのユーザー名とパスワードを入力します。 **[OK]** をクリックします。

5. これで、データフィールドに対してLINQクエリを実行し、必要なデータをエクスポートできるようになりました。たとえば、カタログを右クリックして、 **[Catalogs.Take(100)]** を選択します。これにより、データベースの最初の100のカタログが取得されます。 **[Export]**、 **[Export to Excel with formatting]** の順に選択します。

APIでLINQPadを使用する方法に関する事例については、<http://blogs.citrix.com/2014/01/14/creating-director-custom-reports-for-monitoring-xendesktop/>を参照してください。



# XenAppおよびXenDesktop SDK

Jan 04, 2017

XenAppおよびXenDesktopには、多くのMicrosoft Windows PowerShell Version 3.0スナップインに基づいているSDKが提供されています。これにより、Citrix Studioと同じタスクだけでなく、Studioでサポートされないタスクを実行できます。

Version 7.5以降、XenAppとXenDesktopでは「FlexCast Management Architecture」と呼ばれる共通の統合アーキテクチャが使用されています。これにより、従来XenDesktopでのみ提供されていた多くの機能がXenAppでも提供されるようになりました。このため、これらの機能に関するSDK要素は、XenDesktopについてのみ説明されていても、XenAppとXenDesktopの両方に適用されます。

## XenDesktop 5とXenDesktop 7のSDKの主な違い

- **新しい高レベルのSDK** — XenDesktop 7 SDKでは、サイトの作成や管理を自動化するためのスクリプトを簡単に作成できます。高レベルのSDKにより、複雑な低レベルSDKを使用しなくても、たとえば2つのコマンドレットを実行するだけで新しいサイトを作成できます。
- **新しい低レベルSDK** — XenDesktop 5のBroker SDKに含まれていた、Delegated Administration Service (DAS) 用に拡張されたSDKなど、個々の低レベルSDKが新しいXenDesktop 7サービス用に提供されています。また、Monitor Service、環境テスト、構成ログなどの新しい機能用のSDKもあります。
- **WindowsサーバーOSマシンカタログおよびデリバリーグループ** — XenDesktop 7 SDKを使って、サーバーOS上でホストされるコスト効率のいいアプリケーションやデスクトップを配信できます。
- **デスクトップOSマシンアプリケーション** — デスクトップOSマシンアプリケーションは、SDKレベルで大幅に変更されています。デスクトップOS上でアプリケーションを実行する既存のスクリプトを使用している場合は、後方互換性がほとんどないため、そのスクリプトをXenDesktop 7用に更新する必要があります。
- **デリバリーグループのマシンに対する設定の適用** — XenDesktop 7では、構成スロットを使用して、サイト内のすべてのマシンにではなく、特定のデリバリーグループのマシンに設定を適用できます。これにより、各デリバリーグループに異なる設定を適用できます。ReceiverやApp-V公開サーバーで使用するStoreFrontアドレスの設定など、さまざまな種類の設定を含む多くの定義済み構成スロットが提供されます。スロットの一連の設定を特定のデリバリーグループに適用して、別の設定セットを別のデリバリーグループに適用することができます。これらの設定セットには、「営業部ポリシー」など、わかりやすい名前を付けることができます。
- **カタログの種類置き換え** — XenDesktop 7では、カタログの種類が個々のプロパティを持つカタログに置き換えられています。後方互換性のため、single image (プール) やthin clone (専用) などのカタログの種類を使用した既存のスクリプトも引き続きサポートされますが、内部的にはこれらはプロパティのセットに変換されます。  
注意：XenDesktop 5のカタログの種類との後方互換性は、可能な限り保持されています。ただし、新しいスクリプトを作成する場合はカタログの種類を使用せずに、個々のプロパティを備えたカタログを指定してください。
- **Desktopオブジェクトの置き換え** — XenDesktop 5のBroker SDKスクリプトで使用される主なSDKオブジェクトの1つとして、Desktopオブジェクトがありました。Desktopオブジェクトを使用して、マシン、およびマシン上のセッションの両方を参照できます。XenDesktop 7では、このオブジェクトはSessionオブジェクトとMachineオブジェクトに置き換えられています。どちらもDesktopオブジェクトと同じように動作するよう拡張されています。ただし後方互換性のため、Desktopオブジェクトを使用した既存のスクリプトを引き続きサポートされます。  
注意：XenDesktop 5との後方互換性は、可能な限り保持されています。ただし、新しいスクリプトを作成する場合はDesktopオブジェクトを使用せずに、SessionオブジェクトとMachineオブジェクトを指定してください。

## ポリシー規則での相違

SDKとStudioでは、ポリシー規則の扱い方に違いがあります。SDKでは、ポリシー規則の entitlement (EntitlementPolicyRule) およびassignment (AssignmentPolicyRule) はそれぞれ独立したエンティティです。Studioでは、これらのエンティティはデリバリーグループ内に統合されているため表示されません。また、SDKでのaccessor

リシー規則 (AccessPolicyRule) による制限は、Studioよりも緩いものになっています。

## SDKの使用

このSDKは、ControllerまたはStudioと一緒にインストールされる多くのPowerShellスナップインで構成されています。

コマンドレットにアクセスして実行するには

1. PowerShell 3.0のシェルを開きます。

Studioでシェルを開くには、**[Citrix Studio]** ノードの **[PowerShell]** タブを選択し、**[PowerShellを起動]** をクリックします。

シェルまたはスクリプトを実行するには、Citrix管理者の権限が必要です。Controllerのローカル管理者グループのメンバーにはXenDesktopのインストールに必要な完全な管理権限が自動的に付与されますが、ローカル管理者アカウントを使うのではなく、適切な権限を持つCitrix管理者を作成することをお勧めします。Windows Server 2008を実行している場合、ローカル管理者グループのメンバーとしてではなく、Citrix管理者としてシェルまたはスクリプトを実行する必要があります。

2. スクリプト内でSDKコマンドレットを使用するには、PowerShell実行ポリシーを設定する必要があります。

PowerShell実行ポリシーについて詳しくは、Microsoft社のドキュメントを参照してください。

3. Windows PowerShellコンソールで**Add-PSSnapin**コマンドを使って、必要なスナップインをPowerShell環境に追加します。V1およびV2は、スナップインのバージョンを示します (XenDesktop 5スナップインはV1、XenDesktop 7スナップインはV2)。たとえば、次のコマンドを実行します。

```
Add-PSSnapin Citrix.ADIdentity.Admin.V2
```

すべてのコマンドレットをインポートするには、次のコマンドを実行します。

```
Add-PSSnapin Citrix.*.Admin.V*
```

インポートが完了すると、各コマンドレットとそのヘルプにアクセスできます。

一般的な使用例については、「[SDKの導入](#)」を参照してください。

ヒント：すべてのコマンドレットのヘルプトピックについては、「[PowerShellコマンドレットのヘルプ](#)」を参照してください。

## Group Policy SDKの使用

Citrix Group Policy SDKにより、グループポリシーの設定およびフィルターを表示して構成できます。PowerShellプロバイダーを使用して、マシン、ユーザー設定、およびフィルターに対応する仮想ドライブを作成します。このプロバイダーは、New-PSDriveに対する拡張として表示されます。Group Policy SDKを使用するには、StudioまたはXenApp/XenDesktop SDKのいずれかをインストールする必要があります。

### Group Policy SDKの追加

1. Group Policy SDKを追加するには、次のコマンドを実行します。

```
Add-PSSnapin citrix.common.grouppolicy
```

2. ヘルプにアクセスするには、次のコマンドを実行します。

```
help New-PSDrive -path localgpo:/
```

### Group Policy SDKの使用

1. 仮想ドライブを作成し、設定をロードするには、次のコマンドを実行します。

```
New-PSDrive [-PSProvider] CitrixGroupPolicy
```

```
— -Controller
```

## **New-PSDrive [-PSProvider] CitrixGroupPolicy**

— *-Controller*

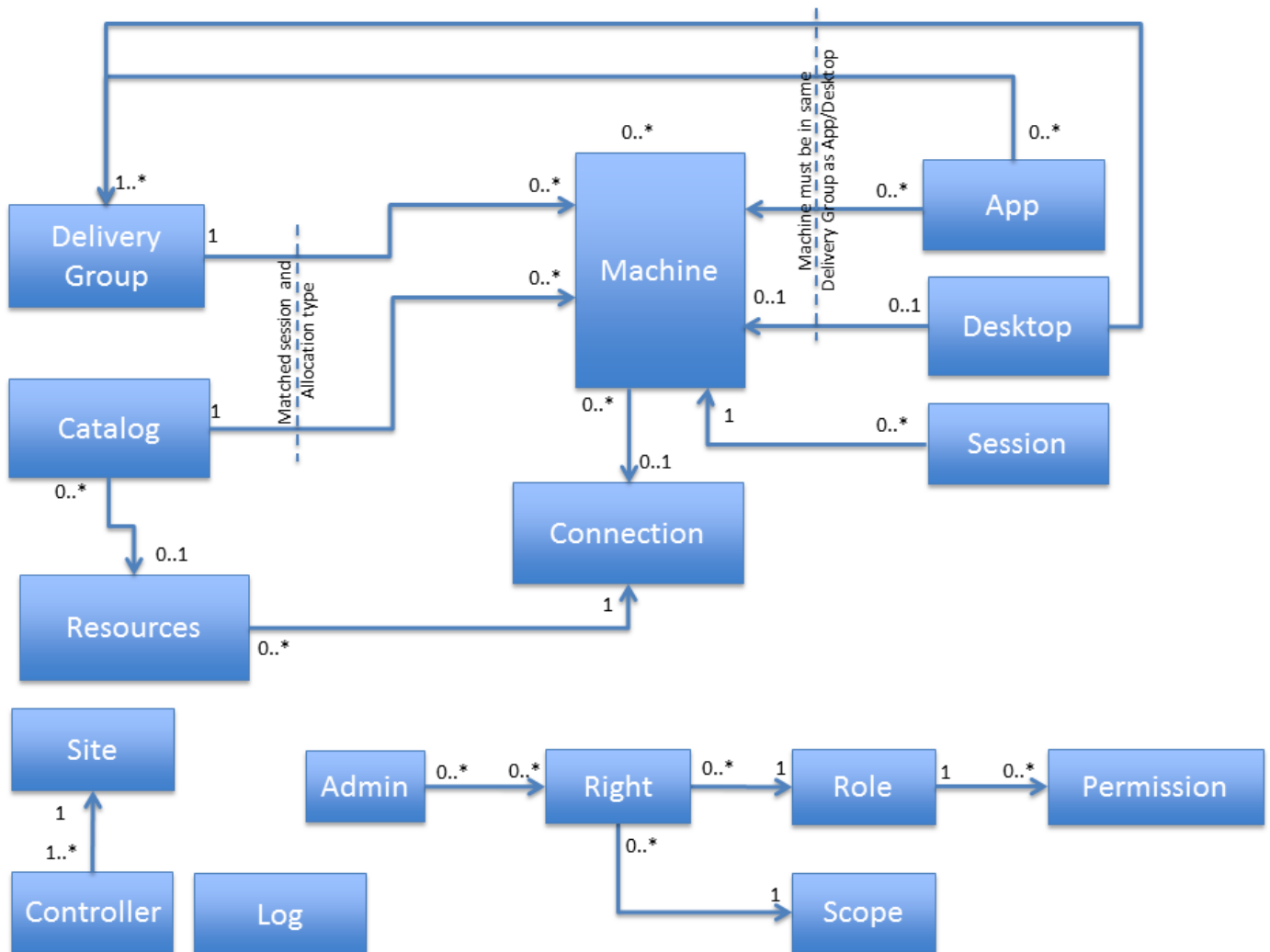
— *-Controller*

でサイトのControllerの完全修飾ドメイン名を指定すると、そのControllerに接続して設定がロードされます。

# XenDesktopの管理モデルについて

Jan 04, 2017

XenDesktop 7の管理モデルは、以下の図に示すように、一連のオブジェクトとこれらのオブジェクトの相互接続で定義されます。



以降のセクションでは、各オブジェクトの機能、特性、および基本的なプロパティについて説明します。

## 接続

Connectionエンティティでは、仮想プラットフォームの管理ポイントへの接続を確立するために必要な詳細が提供されます。この詳細を使用して、必要な電源管理機能とプロビジョニング機能が提供されます。Connectionでは、以下の内容が定義されます。

- 接続のID。名前と内部ID値 (GUID) で構成されます。
- 種類 - XenServer、SCVMM、またはvCenterのいずれか。
- ユーザー名 - ハイパーバイザーへの接続で使用するアカウント名。
- パスワード - ハイパーバイザーへの接続で使用するパスワード。
- 接続の状態、およびハイパーバイザーと通信できるかどうか。
- ハイパーバイザーで発生した現在のアラート。

設定したパスワードが暗号化された形式でデータベースに格納されると、そのパスワードを再読み取りできなくなります。

各プラットフォーム定義のモデルでは、1つのConnectionエンティティがあります。ただし、実行時にはXenDesktop展開環境の要件に対応するために、仮想プラットフォームへの接続が多数確立されます。これらの接続を多数のResourcesエンティティで共有できます（詳しくは次のセクションを参照）。

注：XenServerでConnectionエンティティを設定するときは、ホストまたはXenServerプールを指定します。Microsoft Hyper-VまたはVMwareでConnectionエンティティを設定するときは、管理サーバー（SCVMMまたはvCenter）の詳細を指定します。また、Connectionエンティティを構成する場合、使用する仮想化ソリューションに応じて、ユーザー名を正しい形式で指定する必要があります。XenServerでConnectionエンティティを設定する場合は、ユーザー名のみ（「username」など）を指定します。VMwareまたはHyper-Vで接続エンティティを設定する場合は、ドメインユーザー名（「domain\username」など）を指定します。

## Resources

注：Resourcesエンティティは、Machine Creation Services（MCS）またはProvisioning Services（PVS）を使用して新しいマシンを作成する場合にのみ適用されます。

この設定によって、製品で使用できる仮想化プラットフォーム内のリソースセットが定義されます。これにより、管理者はプラットフォームで提供されるリソースのサブセットに制限したり、目的に応じて異なるリソースセットを定義したりできます。Resourcesエンティティを使用して、サイト内でプロビジョニングされるマシンでのストレージやネットワークの使用を制御できます。Resourcesエンティティでは、以下の内容が定義されます。

- ネットワーク - 仮想化プラットフォームに新しいマシンをプロビジョニングするときに使用できるネットワーク（PVSの場合はストリーム配信ネットワークです。1つのネットワークのみ選択でき、選択しない場合は最初のネットワークが使用されます）。
- ストレージ - 仮想化プラットフォームに新しいマシンをプロビジョニングするときに使用するストレージ。

プロビジョニングメカニズムを使用する場合は、各Connectionオブジェクトに少なくとも1つのResources設定を定義する必要があります。1つのConnectionオブジェクトに多数のResources設定をリンクすることができます。

接続の詳細は、Connectionエンティティによって提供されます。

注：SCVMMまたはvCenterへのConnectionのResourcesを指定する場合は、ストレージとネットワークに加えてホストまたはクラスターを選択します。

## 管理者

Administratorでは、製品の管理者が定義されます。製品の異なる機能セットを使用できる、さまざまな製品管理者を定義できます。Administratorでは、以下の内容が定義されます。

- 管理者を定義するためのADアカウント。管理者の名前とSID値で定義されます。これには、個別のユーザーまたはセキュリティグループを使用できます。
- 管理者の権限（スコープおよび役割）。

## Machine

Machineとは、ユーザーにセッションを提供するためにサイト内で使用できる仮想コンピューターまたは物理コンピューターを指します。Machineでは、以下の内容が定義されます。

- SAM名、DNS名、ホスト（ハイパーバイザー）名、IPアドレス、SID、ライセンスIDなどのマシンID。
- マシンの状態（電源の状態、登録の状態、Personal vDiskの状態、負荷インデックスなど）。これらの状態の多くが1つの状態集計値として集約されます。
- マシンの環境情報と構成情報（インストールされているオペレーティングシステムとCitrixコンポーネントのバージョン番号など。マシンの「機能レベル」を含む）。

- マシンで最後に実行された操作のデータ（最後の登録解除の理由、マシンで実行された最後の電源操作、最後の接続エラーなど）。
- マシンの有効化/無効化/ドレイン動作を制御するメンテナンスモードおよび「WindowsConnectionSetting」の状態。
- 割り当て型のデスクトップの場合、マシンで表示可能なユーザーリソースの設定（アイコン、公開名など）。
- マルチセッションのマシン（サーバーOSマシン）の場合、実行中のセッションの集約情報（アクティブなセッションの数、保留状態のセッションの数など）。
- 割り当てられたマシンに関連付けられた「タグ」値。
- プロビジョニングイメージの情報（マスターイメージのパス、マシンの作成に使用しているプロビジョニングスキーム、次回起動時まで保留されている更新済みイメージがあるかどうかなど）。

ほとんどのマシン値はBroker Service SDKにより公開されますが、アイテムはMachine Creation Services (MCS) などのほかのサービスからも公開されます。マシン単位で示されるほかの値は、該当するマシンが関連付けられているカタログまたはデリバリーグループ（存在する場合）から継承されるか、マシンの作成に使用したプロビジョニングスキームから継承されます。たとえば、マシンの種類は、物理または仮想、静的またはランダム、ユーザーによる変更の保持方法などの要素を組み合わせて、カタログレベルで定義されます。単一セッションのマシン（デスクトップOSマシン）では、該当するマシンで実行されるセッション（存在する場合）の情報もマシンに関連付けられます。この情報には、接続ユーザーのID、セッション状態、使用中のプロトコルなどがあります。Machineは、Catalogの一部として定義されている場合にのみモデル内に存在できます。そうでない場合、Machineは存在できません。

## Catalog

Catalogでは、同等であると認識される（例外もあります）マシンのセットが定義されます。1つの展開環境に複数のCatalogを含めることができ、目的に応じてさまざまなマシンのセットを作成して格納できます。Catalogでは、以下の内容が定義されます。

- カタログの種類。以下のプロパティ値で定義されます。
  1. マシンのプロビジョニング方法（MCS、PVS、または手動）
  2. カタログ内のマシンをユーザーに割り当てる方法（ユーザーに静的かつ永続的に割り当てるのか、各リソース要求時にランダムに割り当てるのか）
  3. 単一セッションのマシン（デスクトップOSマシン）なのかマルチセッションのマシン（サーバーOSマシン）なのか
  4. 物理マシンなのか仮想マシンなのか
  5. ユーザーによる変更内容を保持する方法（ユーザーのログオフ時に変更を破棄するのか、変更をマシン上にローカルに保持するのか、Personal vDiskを使用するのか）
- リモートPCアクセス用のカタログかどうか。リモートPCアクセス用の場合は、カタログに関連付けるActive Directory (AD) OU、およびカタログに関連付けるデリバリーグループ。
- 以下の、MCSのプロビジョニング関連詳細（MCSのプロビジョニングを使用する場合）。
  1. マシンのマスターイメージ
  2. メモリサイズとCPUの数
  3. Personal vDiskのディスクサイズ、ドライブ文字、および割り当て比率
  4. ADアカウントの名前付けスキーム、マシンを作成するOU、および作成済みアカウントの一覧
- そのカタログに関連付けられたPVSサーバーの詳細。
- マシンに想定される機能レベル（これよりも機能レベルが低いマシンはサイトに登録できません）。

また、カタログ内で使用されたマシンや使用可能なマシンについての値もCatalogレベルで公開されます。

## Delivery Group

Delivery Groupでは、エンドユーザーにデスクトップやアプリケーションを提供するために使用されるマシンのグループの詳細が提供されます。複数のDelivery Groupを同じCatalogにリンクして、ユーザーセットの要件に応じて、カタログ内のマシンをさまざまな方法で配信できます。Delivery Groupでは、以下の内容が定義されます。

- デリバリーグループ内のマシンの割り当ての種類。マシンを複数のユーザーで共有するのか（ランダム）1人または複数のユーザーに永続的に割り当てするのか（静的）、およびマシンが単一セッション（デスクトップOS）なのかマルチセッション（サーバーOS）なのかが定義されます。
- デリバリーグループの配信の種類。ユーザーにアプリケーションのみを提供するのか、デスクトップのみを提供するのか、アプリケーションとデスクトップの両方を提供するのかが定義されます。
- デリバリーグループ内のマシンの電源管理の制御設定。以下の設定が含まれます。
  1. 「ピーク」時間と見なされる曜日と時刻
  2. 曜日と時刻に基づいて稼働を維持する、ランダムな未割り当てマシンの数（バッファサイズを含む）
  3. 上記の設定で時刻の評価に使用するタイムゾーン
  4. 割り当てられたマシンの電源管理を行うかどうか。電源管理を行う場合は、その方法
  5. ユーザーの切断やログオフなどのイベントがトリガーされた後で、マシンをシャットダウンまたは一時停止するかどうか。シャットダウンまたは一時停止する場合は、その方法
  6. 実行されたセッションによってマシンが破損したと見なされるかどうか。また、使用後にクリーンな既知の状態に戻すためにマシンを強制的に再起動するかどうか
- デリバリーグループのデスクトップリソースをエンドユーザーに表示する方法（使用するアイコン、色数、名前など）。デリバリーグループ内のマシンのICA接続に必要な、セキュリティレベルも定義されます。デスクトップリソースでは、グループの各ユーザーが同時に使用できるデスクトップの数が定義されます。
- これらのマシンをエンドユーザーが使用できるかどうかを制御するために使用する規則。規則では、ユーザーのIDだけでなく、ユーザーの接続元と接続方法、クライアントデバイスの状態、およびサポートされるリモート処理プロトコルも考えに入れます。
- デリバリーグループを有効にするか無効にするか（メンテナンスモードの設定を含む）。
- エンドユーザーがマシンをリセットできるようにするかどうか（StoreFrontを使用したりリセットなど）。
- ランダム/共有グループに関連付けられた「タグ」値。
- デリバリーグループ内のマシンに想定される機能レベル（これよりも機能レベルが低いマシンは、サイトに登録できません）。
- デリバリーグループをリモートPCアクセスで使用するかどうか、およびデリバリーグループに関連付けられたリモートPCアクセス用のカタログ。
- マルチセッション（サーバーOS）のマシンを特定の曜日と時刻に定期的に再起動するスケジュール、および再起動の方法を制御する設定。

また、Delivery Groupに関連付けられる設定には、プロファイル管理などの機能の設定やStoreFront URLの設定があります。これらの設定は、「Machine Configuration」オブジェクトとして別途定義され、1つまたは複数のDelivery Groupに関連付けられます。

## アプリケーション

Applicationでは、エンドユーザーで使用できるようにする、シームレス（デスクトップから分離された浮動ウィンドウ）アプリケーションの詳細が提供されます。通常、各アプリケーションは1つのデリバリーグループに関連付けられますが、必要に応じて、複数のデリバリーグループでアプリケーション定義を共有することもできます。アプリケーションは、リモートマシンで実行してローカルのクライアントデスクトップに表示したり、ローカルのクライアントマシンにインストールして実行し、そのアプリケーションのウィンドウをリモートデスクトップ上に表示したりできます。Applicationでは、以下の内容が定義されます。

- アプリケーションの種類（「HostedOnDesktop」または「InstalledOnClient」のいずれか）。
- HostedOnDesktopアプリケーションの場合
  1. VDAマシンで実行するアプリケーションの実行可能ファイルのパス、およびアプリケーションの起動時に渡されるコマンドラインパラメーター（指定する場合）。
  2. そのアプリケーションにアクセスできる一連のユーザー（オプション）。デリバリーグループにアクセスできるユーザーのサブセットとして定義。

3. アプリケーションプロセスに適用されるアプリケーション固有の設定（CPU優先レベル、プロキシプリンターの作成を待機するかどうかなど）。
- InstalledOnClientアプリケーションの場合
    1. アプリケーションのアイコンをクライアントデバイスから取得するかどうかを示すフラグ。
    2. アプリケーションに指定されている引数を使用して、追加のセキュリティ機能を適用するかどうかを示すフラグ。
  - アプリケーションリソースをエンドユーザーに表示する方法（使用するアイコン、クライアントデバイス上のフォルダの場所、名前、ショートカットをスタートメニュー、デスクトップ、またはこの両方に表示するかどうかなど）。
  - アプリケーションのファイルタイプの関連付け（ファイル拡張子とアプリケーションの関連付け）。
  - アプリケーションに関連付けられたデリバリーグループ、および複数のグループから選択する場合の優先度値（オプション）。
  - アプリケーションを有効にするかどうか、および（アプリケーションの有効/無効とは別に）エンドユーザーにアプリケーションを表示するかどうか。

## Desktop

XenDesktop 7では、Desktopオブジェクト（マシンおよびマシン上のセッションの両方を示す）は、SessionオブジェクトとMachineオブジェクトに置き換えられており、どちらもDesktopオブジェクトと同じように動作するよう拡張されています。

## セッション

Sessionでは、サイトで制御されるマシンで実行されているWindowsセッションの詳細が提供されます。セッションは、XenDesktopまたはXenAppによって起動されたり、ほかの方法（コンソールやRDPを使ってユーザーがマシンに直接ログオンする場合など）で作成されたりします。

Sessionでは、以下の内容が定義されます。

- セッションを実行しているマシンのID（マシンのDNS名、IPアドレス、NetBios名、SIDなど）。
- セッションを実行しているユーザーのID（SAM名、UPN、SIDなど）。
- セッションへの接続を仲介したユーザーのID（SAM名、SIDなど）。
- セッションへの接続に使用されているエンドポイントクライアントマシンのID（デバイス名、IPアドレス、IDなど）。
- 起動を要求するために使用されたマシン（起動が行われたWebサーバー）のID。このIDには、名前とIPアドレスが含まれます。
- セッションへの接続のゲートウェイとして使用されたマシンのID（マシンのDNS名、IPアドレスなど）。
- セッションでの重要なイベントの詳細（開始時刻、セッションへの接続が最後に行われた時刻、セッションへの仲介が最後に行われた時刻など）。
- 最後のセッション作成またはセッション接続の時間（セッションの仲介にかかった時間、セッションの作成にかかった時間など）。
- セッションの現在の状態（全体的なセッション状態、ICA接続がセキュリティで保護されているかどうか、使用されているプロトコルなど）。
- セッションを実行しているマシンの現在の状態（全体的な状態、電源状態など）。
- セッション接続の確立方法の詳細（セッションの仲介または接続が自律的に行われたのか、セッションコンテキスト「スマートアクセス」タグなどが適用されたのかなど）。
- セッションが「非表示」になっているかどうか。ユーザーがアプリケーションまたはデスクトップを起動したときに特定の種類の問題が発生すると、セッションが非表示になることがあります。
- セッションで実行される仲介アプリケーションの一覧。

Session SDKオブジェクトによって、関連するMachine、Delivery Group、およびCatalogオブジェクトの情報も提供されます。この情報には、ID情報、基本構成、および状態情報が含まれます。

## Controller



Controllerでは、サイト内の個々のDelivery Controllerマシンの詳細が提供されます。そのほとんどは、構成設定ではなく、実行サイトの動的な状態データです。Controllerでは、以下の内容が定義されます。

- 各Controller上で有効になっている「サイトサービス」。
- Controllerコンポーネントのバージョン。
- ControllerマシンのID（完全修飾DNS名、SAM名、SIDなど）。
- Controllerマシンのオペレーティングシステムの種類とバージョン。
- 現在のControllerとサービスの状態、および最後に処理が実行された時刻。
- Controllerに登録されているマシンの数。
- サイトサービスの場所として、Controllerに関連付けられているハイパーバイザー接続。

ほとんどのController値はBroker Service SDKから公開されますが、ほかのアイテムはMachine Creation Servicesなどのほかのサービスから公開されます。

## サイト

Siteとは、XenDesktopサイトの最上位レベルを、そのサイトで実行されている構成サービスの観点から論理的に表したものです。Siteには、ライセンス情報、サイトメタデータ、サイト名などが含まれます。

XenDesktopのインストールには、1つの構成のSiteインスタンスのみが含まれます。Siteオブジェクトには、以下のプロパティがあります。

- 名前
- Controllers - サイト内のControllerの一覧
- Databases - サイトで使用されるデータベースの一覧
- DefaultIconUid
- LicenseInformation
- Metadata

## Log

Logでは、ログ記録が有効になった時点以降にサイトで実行された管理者の操作を示す、収集された構成ログの詳細が提供されます。管理者による読み取り操作はログに記録されませんが、管理者の操作によってサイトの構成や状態が変更される場合、その操作はログに記録されます。高レベルログ、低レベルログ、および操作詳細の3つのレベルのいずれかでログを表示できます。低レベルログは、ログに記録される上位レベル操作の一部です。また、操作詳細には、個々の低レベル操作の要約が示されます。Logアイテムでは、以下の内容が示されます。

- 操作を実行した管理者のID（操作を実行したマシンのIPアドレスなど）。
- 操作を実行した時刻（開始時刻と終了時刻の両方）。
- 操作が成功したか何らかの理由で失敗したか。
- 操作の説明（テキスト文字列。操作の種類、ソースの種類、ターゲットの種類など）。
- 操作で指定したパラメーターの詳細。
- ログ階層の親/子関係。
- 構成変更の場合、アイテムの変更前の値と変更後の値。

ログに記録される内容について詳しくは、[「構成ログ」](#)を参照してください。

## Right

Rightでは、役割と、役割の権限で許可されるスコープの組み合わせが定義されます。役割が指定された管理者は、その役割に定義されている権限を実行できます。ただし、指定されたスコープに直接的または間接的に関連付けられているオブジェク

トに対してのみ実行可能です。Rightでは、以下の内容が定義されます。

- 許可される権限を含んでいる役割。
- 役割の権限が許可されるスコープ。

Rightは個別に操作できるSDKオブジェクトではありません。常に特定のAdministratorオブジェクトに関連付けられます。1人の管理者に複数の権限 (Right) を付与でき、その管理者の管理能力は、付与された権限を合わせたものになります。

## Scope

Scopeではオブジェクトの名前付きグループ (スコープ) が定義され、グループ内のオブジェクトに、役割ごとに制御されるオブジェクトの管理者権限が付与されます。Catalog、Delivery Group、Connection、Resourceなどのさまざまな種類のSDKオブジェクトに、スコープを直接関連付けることができます。これらのSDKオブジェクトのプロパティには、そのオブジェクトが関連付けられているスコープが示されます。ほかのSDKオブジェクトには、スコープが直接関連付けられているほかのオブジェクトとの二次的な関連付けに基づいて、管理者権限が付与されます。たとえば、Machineオブジェクトは、所属しているCatalogとDelivery Groupからスコープの関連付けを継承し、Sessionオブジェクトは、そのセッションを実行するマシンからスコープの関連付けを継承します。一部のスコープ (実際には「すべて」のスコープのみが組み込み型です) は事前定義されていますが、ほかのスコープを作成して、展開環境に適したグループ定義を指定できます。

Scopeでは、以下の内容が定義されます。

- スコープのID。名前と内部ID値 (GUID) で構成されます。
- スコープが組み込み型かどうか。

## 役割

Roleでは、管理者が実行できる権限のセット (役割) が定義されます。Roleは、常に関連付けられたスコープと共に管理者に付与されます。スコープが関連付けられていないオブジェクトに適用される、汎用的な権限を持つ役割もありますが、Role自体によって権限が付与されることはありません。一部の役割は事前定義されていますが、カスタムの役割を作成して、展開環境に適した権限のセットを指定できます。Roleでは、以下の内容が定義されます。

- 役割のID。名前と内部ID値 (GUID) で構成されます。
- 役割が組み込み型かどうか。
- 役割を構成する権限のセット。

組み込みの役割は、次のとおりです。

- すべての管理権限を実行できる管理者 - すべてのタスクと操作を実行できます。
- 読み取り専用管理者 - 全体的な情報に加えて、指定されたスコープのすべてのオブジェクトを表示できますが、変更することはできません。
- マシンカタログ管理者 - マシンカタログを作成および管理したり、マシンをプロビジョニングしたりできます。
- デリバリーグループ管理者 - アプリケーション、デスクトップ、およびマシンを配信したり、それらのセッションを管理したりできます。デリバリーグループやアプリケーションの作成および管理を実行できます。
- ヘルプデスク管理者 - デリバリーグループを表示して、関連付けられているセッションとマシンを管理できます。エンドユーザーの問題のトラブルシューティングを行うために、エンドユーザーリソースの表示や一部の状態変更操作を実行できますが、ほとんどの構成変更は実行できません。
- ホスト管理者 - ホスト接続およびその関連リソース設定を管理できます。

## Permission

Permissionでは、役割に含まれている実行可能な単一のコンソールレベルのタスクまたは操作 (パーミッション) が定義されます。各パーミッションによってさまざまな低レベルSDK操作 (コマンドレット) を実行でき、特定の低レベルSDK操作を継

数のパーミッションに関連付けることもできます。Permissionでは、以下の内容が定義されます。

- パーミッションのID。名前と内部ID値 (GUID) で構成されます。
- パーミッショングループメンバーシップ。パーミッショングループは、特定の機能領域に関連するパーミッションをまとめたものです。
- パーミッションで許可される、一連の低レベルSDK操作。

# SDKの導入

Jan 04, 2017

スクリプトを作成するには、次の手順を実行します。

1. まず最初に、スクリプト化する操作をCitrix Studioを使って実行します。たとえば、Machine Creation Servicesを使用するマシンカタログを作成します。
2. Studioが実行したSDK操作のログを収集します。
3. スクリプトをレビューして、各部の実行内容について把握します。これを理解することで、スクリプトを目的に応じてカスタマイズできるようになります。詳しくは、スクリプトの実行内容についての詳細を説明している使用例を参照してください。
4. Studioスクリプトを必要に応じて編集します。これを行うには、次の操作を行います。
  - 変数を使用します。一部のコマンドレットでは、TaskIDなどのパラメーターが必要です。ただし、Studioは以前のコマンドレットの結果オブジェクトからの値を使用するので、これらのパラメーターで使われる値がどこから来るかは明らかではないことがあります。
  - 不要なコマンドを削除します。
  - 一部の手順をループに追加して、制御しやすいようにできます。たとえば、マシン作成をループに追加すると、作成するマシンの数を制御できます。

## 例

注：SDKに拡張や修正が追加されていることがあるため、例のスクリプトをコピーして貼り付けるのではなく、説明されている手順を実際に実行することをお勧めします。

| 例                                         | 説明                                                     |
|-------------------------------------------|--------------------------------------------------------|
| <a href="#">カタログの作成</a>                   | スクリプト：Machine Creation Services (MCS) のマシン用のカタログを作成します |
| <a href="#">例：ホストの作成と構成</a>               | スクリプト：ホストを作成して構成します                                    |
| <a href="#">例：Personal vDiskデスクトップの作成</a> | スクリプト：Personal vDisk (PvD) デスクトップのデリバリーグループを作成します      |
| <a href="#">例：負荷分散情報の取得</a>               | サーバーOSマシンの負荷インデックス値を表示します                              |

# 例：カタログの作成

Jan 04, 2017

この例では、Machine Creation Services (MCS) のマシン用のカタログを作成します。

まず、「SDKの導入」で説明されている手順を実行しておく必要があります。このドキュメントでは、Studioを使ったスクリプト操作（この場合は、Machine Creation Servicesマシンのセットに対するカタログの作成）の実行方法、およびタスクの実行のためStudioにより実行されたSDK操作のログの収集方法について説明します。これによる出力結果をカスタマイズして、カタログ作成を自動化するスクリプトを作成できます。

注：SDKに拡張や修正が追加されていることがあるため、このドキュメントのスクリプト例をコピーして貼り付けるのではなく、このドキュメントで説明されている手順を実際に行うことをお勧めします。スクリプトを読みやすくするために、行番号と改行が追加されています。

## スクリプトの理解

ここでは、Studioにより生成されるスクリプトの各部で何を実行しているかについて説明します。これを理解することで、スクリプトを目的に応じてカスタマイズできるようになります。読みやすくするために、行番号が追加されています。

1. Start-LogHighLevelOperation -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -Source 'Studio' -StartTime 29/05/2013 14:43:08 -Text 'Create Machine Catalog 'ExampleMachines''  
ログ記録操作を開始してログIDを返します。このログIDは、この後に続くタスクで使用します。

2. New-BrokerCatalog -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -AllocationType 'Permanent' -Description 'Example Machines' -IsRemotePC \$False -LoggingId f39a2792-064a-43eb-97c7-397c  
Brokerのカタログを作成します。このカタログには、作成されるマシンが追加されます。

3. New-AcctIdentityPool -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -AllowUnicode -Domain 'dumdev.internal.citrix.com' -IdentityPoolName 'ExampleMachines' -LoggingId f39a2792-064a-43eb-97c7-397c  
IDプールを作成します。これにより、どのようにADコンピューターアカウントを作成するか定義されます。これは、作成されるマシンに対して作成されるADアカウント用のコンテナになります。

4. Set-BrokerCatalogMetadata -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -CatalogId 1 -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46 -Name 'Citrix\_DesktopStudio\_IdentityPoolUid' -Value  
Brokerカタログで、IDプールの詳細メタデータを設定します。これは必須ではありません。

5. Test-ProvSchemeNameAvailable -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -ProvisioningSchemeName @(ExampleMachines)  
要求した名前が使用可能かどうかをチェックします。これは必須ではありません。

6. New-ProvScheme -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -CleanOnBoot -HostingUnitName 'SharedNFS' -IdentityPoolName 'ExampleMachines' -LoggingId f39a2792-064a-43eb-97c7-397c  
プロビジョニングスキームオブジェクトを作成します。これは作成されるマシンのテンプレートで、ハイパーバイザー、ネットワーク、ストレージ、メモリ、CPUの数などを指定します。また、作られるマシンで使用される仮想マシンスナップショットのバスやホスティングユニット名など、セットアップ済みのパラメーターを取得します。このコマンドでは、仮想マシンスナップショットの「統合された」コピーが作成されます。このため、処理の完了に時間がかかることがあります。

この例では、-RunAsynchronouslyフラグを指定しました。これにより、コマンドが処理を完了する前に管理者に制御が戻るため、この処理が終了するのを待ってから、この処理の完了が必要なほか操作を実行する必要があります。このフラグが指定されてない場合、コマンドはインラインで同期的に実行されます。このため、コマンドが完了するまで制御は戻されません。Get-ProvTaskコマンドレットを使って、非同期的タスクの状態をチェックできます。次のコマンドでは、このタスクを開始した操作（この例ではNew-ProvSchemeコマンドレット）から返されたタスクIDを指定します。

7. Set-BrokerCatalog -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46 -Name 'ExampleMachines' -ProvisioningSchemeId 76125e3a-9001-4993-f  
上記で作成したプロビジョニングスキームの一意なIDで、Brokerカタログを更新します。

8. Add-ProvSchemeControllerAddress -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -ControllerAddress @(DDC.dumdev.internal.citrix.com) -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46 -  
プロビジョニングスキームオブジェクトに一連のControllerのアドレスを追加します。これは、作成されたマシンの展開時にController（ブローカー）に登録するために使用できるアドレスの一覧です。マシンの登録アドレスは、多くの方法で指定できます。ただし、管理者がVDAのインストーラーで [Machine Creation Servicesで自動的に指定する] オプションを指定する場合にはこの情報が必要となります。この一覧に対する変更は、既存のマシンではなく変更の後に作成されたマシンだけに適用されます。

9. Get-AcctADAccount -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -IdentityPoolId b99aee6d-8772-4dbc-978b-8eb9a26e2407 -Lock \$False -MaxRecordCount 2147483647 -State 'Available'  
Studioは、使用できるマシンIDの一覧をIDプールから取得します。これにより、作成済みで未使用のアカウントがある場合は、新しいアカウントを作成する代わりにそのアカウントが使用されます。スクリプトでは代わりに新しいアカウントを作成できるため、既存のアカウントを使用する必要はありません。ただし、スクリプトにアカウントを作成する権限がない場合は、使用可能なアカウントを使用するようにスクリプトを変更してください（このスクリプトを実行する前に、アカウントのプールをIDプールに提供するための処理を実行する必要があります）。

10. New-AcctADAccount -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -Count 2 -IdentityPoolId b99aee6d-8772-4dbc-978b-8eb9a26e2407 -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46  
必要なADコンピューターアカウントをActive Directoryに作成します。このスクリプトでは1つのアカウントが作成されますが、必要な場合はCountパラメーターを使って複数のアカウントを作成できます。アカウントは、上記で作成されたプロビジョニングスキームで定義された組織単位内に作成されます。

11. New-ProvVM -ADAccountName @(DUMDEV\Example-0001\$;DUMDEV\Example-0002\$) -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46  
上記で作成されたプロビジョニングスキームのテンプレート定義に基づいて、仮想マシンを作成します。この処理が完了するまで時間がかかることがあります。

12. Lock-ProvVM -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46 -ProvisioningSchemeName 'ExampleMachines' -Tag 'Brokered' -VMID @(0711  
プロビジョニングされた仮想マシンをロックし、仮想マシンの偶発的な変更を防ぎます。SDKのコンシューマーでは、仮想マシンが使用中でロックされていることを示すためにこれを使用できます。このスクリプトでは、「Brokered」タグで仮想マシンをロックしています。これにより、仮想マシンが作成されてBrokerカタログに追加されていること、およびこのマシンを削除するにはまずこのカタログからマシンを削除する必要があることが示されます。必要に応じて任意のTag名を設定できます。

13. New-BrokerMachine -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -CatalogId 1 -HostedMachineId '0710bb77-d011-d006-4d67-5472e5cd349f' -HypervisorConnectionId 1 -LoggingId f39a2792  
Broker Machineオブジェクトを作成します。これは、プロビジョニングされたマシンとカタログとを結びつける、カタログに格納されたオブジェクトです。

14. Start-BrokerMachinePvdImagePrepare -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -InputObject @(2) -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46  
Personal vDiskのための準備操作の開始をBroker Serviceに要求します。これは、マシンでPersonal vDisk用のストレージを初期化するために必要です。

15. Stop-LogHighLevelOperation -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -HighLevelOperationId f39a2792-064a-43eb-97c7-397cc1238e46 -IsSuccessful \$true  
最初の手順で開始したログ操作を停止して、それが成功したかどうかを示します。

## スクリプトのカスタマイズ

ここでは、Studioからの出力を実際に使用可能なスクリプトとしてカスタマイズする方法について説明します。変数の使用および不要なコマンドの削除に加えて、マシン作成をループに追加して、成されるマシン数を制御しています。読みやすくするために、行番号が追加されています。

```
1. [CmdletBinding()] param ([Parameter(Mandatory=$true)] [string] $hostingUnitPath, [Parameter(Mandatory=$true)] [string] $catalogName, [string] $catalogDescription, [Parameter(Mandatory=$true)] [int])
```

## 例：ホストの作成と構成

Jan 04, 2017

この例では、ホストを作成して構成します。

始める前に、「**SDKの導入**」で説明されている手順に従ってください。これにより、Studioでスクリプト化する操作（この例ではホストの作成）を実行して、StudioによるSDK操作のログを収集します。これによる出力結果をカスタマイズして、ホストの作成を自動化するスクリプトを作成できます。

注：SDKに拡張や修正が追加されていることがあるため、このドキュメントのスクリプト例をコピーして貼り付けるのではなく、このドキュメントで説明されている手順を実際に行うことをお勧めします。スクリプトを読みやすくするために、行番号と改行が追加されています。

### スクリプトの理解

ここでは、Studioにより生成されるスクリプトの各部で何を実行しているかについて説明します。これを理解することで、スクリプトを目的に応じてカスタマイズできるようになります。読みやすくするために、行番号が追加されています。

1. `Get-LogSite -AdminAddress 'mycontroller.example.com:80'`  
構成ログサービスを照会して、サイトの構成についての情報を取得します。
2. `Start-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -Source 'Studio' -StartTime 14/08/2013 14:30:28 -Text 'Create Connection "Example XenServer"'`  
ハイレベルの構成ログ操作を開始します。このログに以降の手順のコマンドが記録されます。これにより、以降の操作で使用するログID (LoggingId) が返されます。
3. `Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'`  
Host Serviceの場所を設定します。この場所は、構成コマンドレットで使用します。Host Serviceは単一PowerShellプロバイダーを提供するため、すべてのコマンドレットでこのサービスのアドレスを使用できるわけではありません。このコマンドレットでは、デフォルトの場所を設定します。
4. `New-Item -ConnectionType 'XenServer' -HypervisorAddress @( 'http://xenhost1.example.com' ) -LoggingId e355ce51-8cbb-400a-ae81-1fdc567239cb -Path @( 'XDHyp:\Connections\Example XenServer XenServer (xenhost1.example.com)' )`  
この接続は非永続的で、このPowerShell実行空間でのみ有効です。
5. `Stop-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -EndTime 14/08/2013 14:30:29 -HighLevelOperationId 'e355ce51-8cbb-400a-ae81-1fdc567239cb' -IsSuccessful $True`  
開始したログ操作を停止して、それが成功したかどうかを示します。
6. `Get-LogSite -AdminAddress 'mycontroller.example.com:80'`  
構成ログサービスを照会して、サイトの構成についての情報を取得します。
7. `Start-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -Source 'Studio' -StartTime 14/08/2013 14:30:30 -Text 'Update Connection "Example XenServer"'`  
新しいハイレベルログ操作を開始します。
8. `Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'`  
Host Serviceの詳細を再度設定します（この反復は下記のカスタマイズ済みのスクリプトでは削除されています）。
9. `Set-Item -HypervisorAddress @( 'http://xenhost1.example.com', 'http://xenhost2.example.com' ) -LoggingId 44e15629-6906-4840-a36c-984aaf67be6d -PassThru -Path @( 'XDHyp:\Connections\Example XenServer (xenhost1.example.com)', 'XDHyp:\Connections\Example XenServer (xenhost2.example.com)' )`  
手順4.で作成した接続を更新します。プール内に複数のXenServerホストが存在するため、すべてのアドレスを指定して高可用性機能を有効にします。
10. `Stop-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -EndTime 14/08/2013 14:30:31 -HighLevelOperationId '44e15629-6906-4840-a36c-984aaf67be6d' -IsSuccessful $True`  
手順7.で開始したログ操作を停止します。
11. `Get-LogSite -AdminAddress 'mycontroller.example.com:80'`  
構成ログサービスを照会して、サイトの構成についての情報を取得します。
12. `Start-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -Source 'Studio' -StartTime 14/08/2013 14:31:03 -Text 'Create Resources "Example Resources" and Persist Connection "Example XenServer"'`  
新しいログ操作を開始します。
13. `Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'`  
Host Serviceの詳細を再度設定します。
14. `Get-ChildItem -Path @( 'XDHyp:\Connections' )`  
ウィザードでの入力が必要なホスト接続の情報を取得します。
15. `Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'`  
Host Serviceの詳細を再度設定します。
16. `Remove-Item -LoggingId 76caa3f4-df93-4cb2-b78d-6a8824766314 -Path @( 'XDHyp:\Connections\Example XenServer (xenhost1.example.com)' )`  
ウィザードで作成された一時的な接続を削除します。
17. `Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'`  
Host Serviceの詳細を再度設定します。
18. `New-Item -ConnectionType 'XenServer' -HypervisorAddress @( 'http://xenhost1.example.com', 'http://xenhost2.example.com' ) -LoggingId 76caa3f4-df93-4cb2-b78d-6a8824766314 -Path @( 'XDHyp:\Connections\Example XenServer (xenhost1.example.com)', 'XDHyp:\Connections\Example XenServer (xenhost2.example.com)' )`  
永続的な新しい接続を作成します。この接続はデータベースに書き込まれ、ほかのPowerShell実行空間で使用できるようになります。
19. `New-BrokerHypervisorConnection -AdminAddress 'mycontroller.example.com:80' -HypHypervisorConnectionUid a14096ba-5074-44ff-b596-371e345c0449 -LoggingId 76caa3f4-df93-4cb2-b78d-6a8824766314`  
このホスト接続をBroker Serviceに追加します。
20. `Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'`  
Host Serviceの詳細を再度設定します。
21. `New-Item -HypervisorConnectionName 'Example XenServer' -LoggingId 76caa3f4-df93-4cb2-b78d-6a8824766314 -NetworkPath @( 'XDHyp:\Connections\Example XenServer\Network 0.network' ) -ParentPath @( 'XDHyp:\Connections\Example XenServer' )`  
手順14.で取得した情報を使用して、HostingUnit (Studioで「リソース」と呼ばれるもの) を作成します。
22. `Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'`  
Host Serviceの詳細を再度設定します。
23. `Get-Item -Path @( 'XDHyp:\Connections\Example XenServer' )`  
新たに作成されたオブジェクトを取得します。

Stop-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -EndTime 14/08/2013 14:31:07 -HighLevelOperationId '76caa3f4-df93-4cb2-b78d-6a8824766314' -IsSuccessful \$True  
開始したログ操作を停止して、それが成功したかどうかを示します。

#### スクリプトのカスタマイズ

ここでは、Studioからの出力を実際に使用可能なスクリプトとしてカスタマイズする方法について説明します。上記の手順ではウィザードに必要な情報を取得するために一時的なホスト接続を作成しましたが、以下のスクリプトでは1つの永続的な接続のみを作成します。この接続内で情報を取得して、HostingUnit (リソース) を作成します。ここで、LoggingIdとHypHyperConnectionUidの詳細が異なることに注意してください。

読みやすくするために、行番号が追加されています。各行が個別のPowerShellコマンドです。

1. Start-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -Source 'Studio' -Text 'Create Connection `Example XenServer`" 2.Set-HypAdminConnection -AdminAddress 'mycontrolle

# 例：Personal vDiskデスクトップの作成

Jan 04, 2017

次の例では、Personal vDisk (PvD) デスクトップ用のデリバリーグループを作成します。

始める前に、「**SDKの導入**」で説明されている手順に従ってください。これにより、Studioでスクリプト化する操作を実行して、StudioによるSDK操作のログを収集します。これによる出力結果をカスタマイズして、このタスクを自動化するスクリプトを作成できます。

注：SDKに拡張や修正が追加されていることがあるため、このドキュメントのスクリプト例をコピーして貼り付けるのではなく、このドキュメントで説明されている手順を実際に行うことをお勧めします。

## スクリプトの理解

ここでは、Studioにより生成されるスクリプトの各部で何を実行しているかについて説明します。これを理解することで、スクリプトを目的に応じてカスタマイズできるようになります。スクリプトを読みやすくするために、行番号と改行が追加されています。

1. Start-LogHighLevelOperation -AdminAddress 'test-ddc.mydomain.com:80' -Source 'Studio' -StartTime 31/07/2013 10:08:58 -Text 'Create Delivery Group `Win7 PvD Desktops`'  
ログ記録操作を開始してログIDを返します。このログIDは、この後に続くタスクで使用します。
2. New-BrokerDesktopGroup -AdminAddress 'test-ddc.mydomain.com:80' -ColorDepth 'TwentyFourBit' -DeliveryType 'DesktopsOnly' -DesktopKind 'Private' -InMaintenanceMode \$False -IsRemotePC \$False  
Studioのウィザードにより収集されたオプションを持つ新しいデリバリーグループを作成します。
3. Add-BrokerMachinesToDesktopGroup -AdminAddress 'test-ddc.mydomain.com:80' -Catalog 'win7-pvd' -Count 2 -DesktopGroup 'Win7 PvD Desktops' -LoggingId 846f2d42-a994-4bce-ab58-be05c8d73b99  
新しいデリバリーグループに、指定のカタログから要求されたマシンを追加します。
4. Set-Variable -Name 'brokerUsers' -Value @( 'S-1-5-21-3291547628-200264090-930806513-1104'; 'S-1-5-21-3291547628-200264090-930806513-1105' ) Get-BrokerUser -AdminAddress 'test-ddc.mydomain.com:80'  
上記のコマンドは必須ではなく、Studioがユーザーを検証しています。
5. Test-BrokerAssignmentPolicyRuleNameAvailable -AdminAddress 'test-ddc.mydomain.com:80' -Name @( 'Win7 PvD Desktops' )  
Studioは、ポリシー割り当て名が使用できるかどうかチェックします。
6. New-BrokerAssignmentPolicyRule -AdminAddress 'test-ddc.mydomain.com:80' -DesktopGroupUid 41 -Enabled \$True -IncludedUserFilterEnabled \$False -LoggingId 846f2d42-a994-4bce-ab58-be05c8d73b99  
デリバリーグループに対する新しいポリシー割り当て規則を作成します。ここではユーザーを指定しないため、すべての制御がアクセスポリシー規則を介して行われます。
7. Set-Variable -Name 'brokerUsers' -Value @( 'S-1-5-21-3291547628-200264090-930806513-1104'; 'S-1-5-21-3291547628-200264090-930806513-1105' ) Get-BrokerUser -AdminAddress 'test-ddc.mydomain.com:80'  
上記のコマンドは必須ではなく、Studioがより詳細なチェックを実行しています。
8. Test-BrokerAccessPolicyRuleNameAvailable -AdminAddress 'test-ddc.mydomain.com:80' -Name @( 'Win7 PvD Desktops\_Direct' )  
Studioは、アクセスポリシー規則名が使用できるかどうかチェックします。
9. New-BrokerAccessPolicyRule -AdminAddress 'test-ddc.mydomain.com:80' -AllowedConnections 'NotViaAG' -AllowedProtocols @( 'HDX'; 'RDP' ) -AllowRestart \$True -DesktopGroupUid 41 -Enabled \$True  
非NetScaler Gateway接続の新しいデスクトップに対してアクセスポリシー規則を作成します。
10. Test-BrokerAccessPolicyRuleNameAvailable -AdminAddress 'test-ddc.mydomain.com:80' -Name @( 'Win7 PvD Desktops\_AG' ) New-BrokerAccessPolicyRule -AdminAddress 'test-ddc.mydomain.com:80'  
Studioはこの処理をNetScaler Gateway接続に対しても実行します。
11. Test-BrokerPowerTimeSchemeNameAvailable -AdminAddress 'test-ddc.mydomain.com:80' -Name @( 'Win7 PvD Desktops\_Weekdays' ) New-BrokerPowerTimeScheme -AdminAddress 'test-ddc.mydomain.com:80'  
Studioは、平日および週末の電源スキーム（オプション）の名前を使用できるかどうかをチェックして、それらを追加します。
12. Stop-LogHighLevelOperation -AdminAddress 'test-ddc.mydomain.com:80' -EndTime 31/07/2013 10:09:05 -HighLevelOperationId '846f2d42-a994-4bce-ab58-be05c8d73b99' -IsSuccessful \$True  
手順1で開始したログ操作を停止して、それが成功したかどうかを示します。

## スクリプトのカスタマイズ

ここでは、Studioからの出力を実際に使用可能なスクリプトとしてカスタマイズする方法について説明します。

Personal vDiskデスクトップ用のデリバリーグループを作成します。パラメーターでは、静的な割り当て種類およびPvDディスクが設定された既存のマシンカタログを指定する必要があります。このスクリプトを実行するには、Citrix管理者としてログオンしたPowershellコマンドラインを使用する必要があります。権限に対するチェックは実行されません。そのため、ユーザーに適切な権限がない場合にはスクリプトの実行に失敗します。

<# Sample usage: .\CreatePvDGroup.ps1 -GroupName "Win7 PvD Desktops" -SrcCatalog "win7-pvd" -NumDesktops 2 -Users @( 'mydomain\user1'; 'mydomain\user2' ) -#> Param ( [Parameter(Mandatory)] \$Group, [Parameter(Mandatory)] \$Catalog, [Parameter(Mandatory)] \$Count, [Parameter(Mandatory)] \$Users )

| パラメーター      | 説明                                                                                                        |
|-------------|-----------------------------------------------------------------------------------------------------------|
| GroupName   | 新しいPvDデスクトップグループの名前。                                                                                      |
| SrcCatalog  | PvDデスクトップの作成に使用されるカタログの名前。割り当ての種類に静的を指定してカタログを作成します。マシンには、PvDディスクもある必要があります。                              |
| NumDesktops | PvDデスクトップグループに追加するマシンの数。十分な数のマシンがない場合は、使用可能な最大数のマシンが追加されます。                                               |
| ユーザー        | グループにアクセスできるユーザー。「@(mydomain\Domain Users)」や「@(mydomain\user1;mydomain\user2)」など、ユーザーまたはグループのリストとして指定します。 |

Set-HypAdminConnection -AdminAddress \$adminAddress

使用するハイパーバイザー管理接続を指定します。これにより、一部のコマンドで-AdminAddressの指定が不要になります。

\$peakPoolSize = 2 \$weekendPoolSizeByHour = new-object int[] 24 \$weekdayPoolSizeByHour = new-object int[] 24 9..17 | %{ \$weekdayPoolSizeByHour[\$\_] = \$peakPoolSize } \$peakHours = (0..23 | %{ \$weekdayPoolSizeByHour[\$\_] })  
これにより、各エントリに1または0がある24要素の配列が作成されます。これを使って、デリバリーグループの電源スケジュールのピーク時がいつかを指定します。平日の要素9~17 (09:00~17:00) を1に、そのほかの時間については0のままで設定します。可能な場合、ピーク時間中に2つの未割り当てマシンの電源がオンになります。



```
$logId = Start-LogHighLevelOperation -Text "Create PvD desktop group" -Source "Create PvD Desktop Group Script"
```

新しいログ操作を開始します。これにより、次のグループの作成タスクで使用するログIDが返されます。

```
$grp = New-BrokerDesktopGroup -DesktopKind 'Private' -DeliveryType 'DesktopsOnly' -LoggingId $logId -Name $GroupName -PublishedName $GroupName -SessionSupport 'SingleSession'
```

プライベートデスクトップを配信する新しいデリバリーグループを作成します。使用されるカタログには、適切なマシン（PvDディスクを持つ永続マシン）が設定されている必要があります。PublishedNameはエンドユーザーに表示される名称です。以下では、グループ名と同じ名前が使用されます。

```
New-BrokerAssignmentPolicyRule -DesktopGroupUid $grp.Uid -IncludedUserFilterEnabled $False -LoggingId $logId -MaxDesktops 1 -Name ($GroupName + '_AssignRule') | Out-Null
```

割り当てられたデスクトップは割り当てポリシーを必要とします。ユーザーフィルターを無効にして、アクセスがすべてアクセスポリシー規則で制御されるようにします。

```
New-BrokerAccessPolicyRule -AllowedConnections 'NotViaAG' -AllowedProtocols @('HDX','RDP') -AllowRestart $True -DesktopGroupUid $grp.Uid -IncludedSmartAccessFilterEnabled $True -Is
```

アクセス制限を指定します。NetScaler Gatewayを使った直接アクセス、HDXおよびRDPプロトコルを使った直接アクセスを許可します。必要に応じて、ユーザーはデスクトップの再起動を要求できます。

```
New-BrokerPowerTimeScheme -DaysOfWeek 'Weekdays' -DesktopGroupUid $grp.Uid -DisplayName 'Weekdays' -LoggingId $logId -Name ($GroupName + '_Weekdays') -PeakHours $peakHours
```

必要な場合は、電源スケジュールを指定します。

```
Stop-LogHighLevelOperation -HighLevelOperationId $logId -IsSuccessful $True
```

構成ログを停止して、成功したかどうかを示します。

# 例：負荷分散情報の取得

Jan 04, 2017

サーバーOSマシンでは、サーバーOS上でホストされるコスト効率に優れたアプリケーションやデスクトップを複数のユーザーに配信できます。

展開環境内のサーバーOSマシンの負荷を分散するには、Citrixポリシーを使用します。Citrixポリシーには、WindowsサーバーOSマシンを配信するサーバー間で負荷管理を有効にして構成するための複数の設定項目があります。詳しくは、負荷管理の設定項目に関するリファレンスドキュメントを参照してください。ポリシーを設定するには、Studio、またはWindowsのグループポリシー管理コンソールを使用します。詳しくは、ポリシーのドキュメントを参照してください。

負荷を確認するには、Citrix Directorコンソール、Studioコンソール、またはPowerShell SDKのいずれかを使用できます。次の例では、PowerShell SDKを使用して負荷情報を表示します。

注：以前のバージョンのXenDesktopでは、`qfarm /load`コマンドを使用して負荷情報を確認できました。このバージョンではこのコマンドを使用することはできませんが、PowerShellを使って同様の情報を表示することができます。

## 例：負荷インデックス値の取得

次のコマンドを実行すると、各マシンで実行中のセッション数と測定された負荷インデックス値の一覧が表示されます。

1. PowerShellのシェルを開始します。詳しくは、「[SDK](#)」を参照してください。
2. 以下のコマンドを入力します。

```
Get-BrokerMachine -SessionSupport MultiSession -Property 'DnsName','LoadIndex','SessionCount'
```

注：負荷インデックス値は、最大10000です。負荷インデックス値はセッション数などの測定基準で計算され、VDAマシンの負荷を示します。値が10000の場合、VDAマシンの負荷が限界状態であることを示します。この場合、ブローカーはそれ以上のユーザーセッションをそのマシンに割り当てなくなります。

追加の情報および例については、「PowerShellコマンドレットのヘルプ」の「`Get-BrokerMachine`」や「`about_Broker_Filtering`」などのトピックに記載されています。「[PowerShellコマンドレットのヘルプ](#)」を参照してください。

# PowerShellコマンドレットのヘルプ

Jan 04, 2017

ここでは、すべてのPowerShellコマンドレットに関するヘルプテキストを参照できます。

- Citrix.AdIdentity.Admin.V2
- Citrix.AppV.Admin.V1
- Citrix.Broker.Admin.V2
- Citrix.Configuration.Admin.V2
- Citrix.ConfigurationLogging.Admin.V1
- Citrix.DelegatedAdmin.Admin.V1
- Citrix.EnvTest.Admin.V1
- Citrix.Host.Admin.V2
- Citrix.MachineCreation.Admin.V2
- Citrix.Monitor.Admin.V1
- Citrix.Storefront.Admin.V1

# Citrix.AdIdentity.Admin.V2

Jan 04, 2017  
Overview

| Name                                 | Description                                                       |
|--------------------------------------|-------------------------------------------------------------------|
| <a href="#">AcctAdIdentitySnapin</a> | The Active Directory Identity Service PowerShell snap-in provides |
| <a href="#">Acct Filtering</a>       | Describes the common filtering options for XenDesktop cmdlets.    |

## Cmdlets

| Name                                           | Description                                                                                         |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <a href="#">Add-AcctADAccount</a>              | Import Active Directory computer accounts from Active Directory for use in the AD Identity Service. |
| <a href="#">Add-AcctIdentityPoolScope</a>      | Add the specified IdentityPool(s) to the given scope(s).                                            |
| <a href="#">Copy-AcctIdentityPool</a>          | Copies an Identity Pool and its associated Identities to a new IdentityPool                         |
| <a href="#">Get-AcctADAccount</a>              | Gets the AD accounts stored in the AD Identity Service.                                             |
| <a href="#">Get-AcctDBConnection</a>           | Gets the database string for the specified data store used by the AdIdentity Service.               |
| <a href="#">Get-AcctDBSchema</a>               | Gets a script that creates the AdIdentity Service database schema for the specified data store.     |
| <a href="#">Get-AcctDBVersionChangeScript</a>  | Gets a script that updates the AdIdentity Service database schema.                                  |
| <a href="#">Get-AcctIdentityPool</a>           | Gets existing identity pools.                                                                       |
| <a href="#">Get-AcctInstalledDBVersion</a>     | Gets a list of all available database schema versions for the AdIdentity Service.                   |
| <a href="#">Get-AcctScopedObject</a>           | Gets the details of the scoped objects for the AdIdentity Service.                                  |
| <a href="#">Get-AcctService</a>                | Gets the service record entries for the AdIdentity Service.                                         |
| <a href="#">Get-AcctServiceAddedCapability</a> | Gets any added capabilities for the AdIdentity Service on the controller.                           |

| Name                               | Description                                                                                    |
|------------------------------------|------------------------------------------------------------------------------------------------|
| Get-AcctServiceInstance            | Gets the service instance entries for the AdIdentity Service.                                  |
| Get-AcctServiceStatus              | Gets the current status of the AdIdentity Service on the controller.                           |
| New-AcctADAccount                  | Creates AD computer accounts in the specified identity pool.                                   |
| New-AcctIdentityPool               | Creates a new identity pool.                                                                   |
| Remove-AcctADAccount               | Removes AD computer accounts from an identity pool.                                            |
| Remove-AcctIdentityPool            | Removes identity pools.                                                                        |
| Remove-AcctIdentityPoolMetadata    | Removes metadata from the given IdentityPool.                                                  |
| Remove-AcctIdentityPoolScope       | Remove the specified IdentityPool(s) from the given scope(s).                                  |
| Remove-AcctServiceMetadata         | Removes metadata from the given Service.                                                       |
| Rename-AcctIdentityPool            | Renames an identity pool.                                                                      |
| Repair-AcctADAccount               | Resets the Active Directory machine password for the given accounts.                           |
| Reset-AcctServiceGroupMembership   | Reloads the access permissions and configuration service locations for the AdIdentity Service. |
| Set-AcctDBConnection               | Configures a database connection for the AdIdentity Service.                                   |
| Set-AcctIdentityPool               | Update parameters of an identity pool.                                                         |
| Set-AcctIdentityPoolMetadata       | Adds or updates metadata on the given IdentityPool.                                            |
| Set-AcctServiceMetadata            | Adds or updates metadata on the given Service.                                                 |
| Test-AcctDBConnection              | Tests a database connection for the AdIdentity Service.                                        |
| Test-AcctIdentityPoolNameAvailable | Checks to ensure that the proposed name for an identity pool is unused.                        |
| Unlock-AcctADAccount               | Unlocks AD accounts within the AD Identity Service.                                            |

| <b>Name</b>                             | <b>Description</b>                                                         |
|-----------------------------------------|----------------------------------------------------------------------------|
| <a href="#">Unlock-AcctIdentityPool</a> | Unlocks Identity pools.                                                    |
| <a href="#">Update-AcctADAccount</a>    | Refreshes the AD computer account state stored in the AD Identity Service. |

# about\_AcctAdIdentitySnapin

Jan 04, 2017

## TOPIC

about\_AcctAdIdentityServiceSnapin

## SHORT DESCRIPTION

The Active Directory Identity Service PowerShell snap-in provides administrative functions for the Active Directory Identity Service.

## COMMAND PREFIX

All commands in this snap-in have 'Acct' in their name.

## LONG DESCRIPTION

The Active Directory Identity Service PowerShell snap-in enables both local and remote administration of the Active Directory Identity Service. It provides facilities to store details about Active Directory computer accounts that the Machine Creation Service can use.

The snap-in provides two main entities:

### Identity

A representation of an Active Directory computer account that reflects the state of the account within the context of the Machine Creation Service. When an account is created by or imported into the Active Directory Identity Service, the account password is stored. Once the account is consumed by the Machine Creation Service, the password is discarded. For accounts registered with the Active Directory Identity Service, identities hold the following additional state information.

#### Available

The Active Directory account is registered with the service, the password for the account is known, and the account is available to be consumed by another service. Accounts that are successfully created with the `New-AcctADAccount` command or imported using the `Add-ADAccount` command, are initially assigned this state.

#### InUse

The Active Directory account is registered and has been consumed by another service. The password for the account is no longer known to the service.

#### Error

The Active Directory account is registered, but is missing,

disabled, or locked within Active Directory. Accounts that are not successfully created with the `New-AcctADAccount` command or imported using the `Add-ADAccount` command appear in this state. Use the `Update-AcctADAccount` and `Repair-AcctADAccount` commands to resolve issues with accounts in this state.

#### Tainted

The Active Directory account is registered and has been released by all the consuming services, but cannot be made available for use as the password is no longer known. Use the `Repair-AcctADAccount` command to reset account passwords and restore the account state to 'Available'.

Identities can also be marked as 'Locked' by the Machine Creation Service to indicate that they are in use and must not be changed. These services are also responsible for unlocking the Active Directory accounts when they no longer require exclusive access. Use the `Unlock-AcctADAccount` command to allow the lock to be overridden, if necessary.

#### Identity Pool

Containers for identities that can be configured with all the information required for new Active Directory accounts to be created. Alternatively, identity pools can be populated by importing accounts that already exist in Active Directory. All accounts registered with the Active Directory Identity Service must be placed into one of these containers. An identity can belong to more than one identity pool, but the state of the identity cannot be different in each pool. For example, an identity that is in use will be marked 'InUse' in all the identity pools of which it is part.

To avoid conflicting changes, identity pools can also be marked as 'Locked'

during operations that modify the content of a pool. These

operations are also responsible for unlocking the identity pool. Use the `unlock-AcctIdentityPool` command to allow the lock to be overridden, if necessary.

## ACTIVE DIRECTORY PERMISSIONS

### Account Creation (using the `New-AcctADAccount` command)

To use PowerShell to create new Active Directory accounts, the `runspace`



must be run using an account with sufficient permissions in the required Active Directory container (specified by the identity pool organizational unit parameter) for accounts to be created.

Import Accounts (using the Add-AcctADAccount command)

There are two modes for this operation: situations where the Active Directory account passwords are known and situations where the passwords are not known.

If the account passwords are known, the accounts can be imported without the need for administrative permissions in Active Directory. The accounts are imported and the password provided is used to change the existing password.

If the passwords are not known, the runspace must be run using an account that has permissions to reset the password for the accounts.

# about\_Acct\_Filtering

Jan 04, 2017

## TOPIC

XenDesktop - Advanced Dataset Filtering

## SHORT DESCRIPTION

Describes the common filtering options for XenDesktop cmdlets.

## LONG DESCRIPTION

Some cmdlets operate on large quantities of data and, to reduce the overhead of sending all of that data over the network, many of the Get- cmdlets support server-side filtering of the results.

The conventional way of filtering results in PowerShell is to pipeline them into Where-Object, Select-Object, and Sort-Object, for example:

```
Get-<Noun> | Where { $_.Size = 'Small' } | Sort 'Date' | Select -First 10
```

However, for most XenDesktop cmdlets the data is stored remotely and it would be slow and inefficient to retrieve large amounts of data over the network and then discard most of it. Instead, many of the Get- cmdlets provide filtering parameters that allow results to be processed on the server, returning only the required results.

You can filter results by most object properties using parameters derived from the property name. You can also sort results or limit them to a specified number of records:

```
Get-<Noun> -Size 'Small' -SortBy 'Date' -MaxRecordCount 10
```

You can express more complex filter conditions using a syntax and set of operators very similar to those used by PowerShell expressions.

Those cmdlets that support filtering have the following common parameters:

-MaxRecordCount <int>

Specifies the maximum number of results to return.  
For example, to return only the first nine results use:

```
Get-<Noun> -MaxRecordCount 9
```

If not specified, only the first 250 records are returned, and if more are available, a warning is produced:

WARNING: Only first 250 records returned. Use -MaxRecordCount to retrieve more.

You can suppress this warning by using -WarningAction or by specifying a value for -MaxRecordCount.

To retrieve all records, specify a large number for -MaxRecordCount. As the value is an integer, you can use the following:

```
Get-<Noun> -MaxRecordCount [int]::MaxValue
```

-ReturnTotalRecordCount [<SwitchParameter>]

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. For example:

```
Get-<Noun> -MaxRecordCount 9 -ReturnTotalRecordCount
```

....

```
Get-<Noun> : Returned 9 of 10 items
```

```
At line:1 char:18
```

```
+ Get-<Noun> <<<< -MaxRecordCount 9 -ReturnTotalRecordCount
```

```
+ CategoryInfo          : OperationStopped: (:) [Get-<Noun>], PartialDataException
```

```
+ FullyQualifiedErrorId : PartialData,Citrix.<SDKName>.SDK.Get<Noun>
```

The count can be accessed using the TotalAvailableResultCount property:

```
$count = $error[0].TotalAvailableResultCount
```

-Skip <int>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

-SortBy <string>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order, respectively. Ascending order is assumed if no prefix is present.

Sorting occurs before -MaxRecordCount and -Skip parameters are applied. For example, to sort by Name and then by Count (largest first) use:

```
-SortBy 'Name,-Count'
```

By default, sorting by an enumeration property uses the numeric value of the elements. You can specify a different sort order by qualifying the name with an ordered list of elements or their numeric values, or <null> to indicate the placement of null values.

Elements not mentioned are placed at the end in their numeric order. For example, to sort by two different enums and then by the object id:

```
-SortBy 'MyState(StateC,<null>,StateA,StateB),Another(0,3,2,1),Id'
```

-Filter <String>

This parameter lets you specify advanced filter expressions, and supports combination of conditions with -and and -or, and grouping with braces. For example:

```
Get-<Noun> -Filter 'Name -like "High*" -or (Priority -eq 1 -and Severity -ge 2)'
```

The syntax is close enough to PowerShell syntax that you can use script blocks in most cases. This can be easier to read as it reduces quoting:

```
Get-<Noun> -Filter { Count -ne $null }
```

The full -Filter syntax is provided below.

## EXAMPLES

Filtering by strings performs a case-insensitive wildcard match.

Separate parameters are combined with an implicit -and operator.

Normal PowerShell quoting rules apply, so you can use single or double quotes, and omit the quotes altogether for many strings. The order of parameters does not make any difference. The following are equivalent:

```
Get-<Noun> -Company Citrix -Product Xen*
Get-<Noun> -Company "citrix" -Product '[X]EN*'
Get-<Noun> -Product "Xen*" -Company "CITRIX"
Get-<Noun> -Filter { Company -eq 'Citrix' -and Product -like 'Xen*' }
```

See [about\\_Quoting\\_Rules](#) and [about\\_Wildcards](#) for details about PowerShell handling of quotes and wildcards.

To avoid wildcard matching or include quote characters, you can escape the wildcards using the normal PowerShell escape mechanisms (see [about\\_Escape\\_Characters](#)), or switch to a filter expression and the `-eq` operator:

```
Get-<Noun> -Company "Abc[*]"           # Matches Abc*
Get-<Noun> -Company "Abc`*"           # Matches Abc*
Get-<Noun> -Filter { Company -eq "Abc*" } # Matches Abc*
Get-<Noun> -Filter { Company -eq "A`"B`C" } # Matches A"B'C
```

Simple filtering by numbers, booleans, and TimeSpans perform direct equality comparisons, although if the value is nullable you can also search for null values. Here are some examples:

```
Get-<Noun> -Uid 123
Get-<Noun> -Enabled $true
Get-<Noun> -Duration 1:30:40
Get-<Noun> -NullableProperty $null
```

More comparisons are possible using advanced filtering with `-Filter`:

```
Get-<Noun> -Filter 'Capacity -ge 10gb'
Get-<Noun> -Filter 'Age -ge 20 -and Age -lt 40'
Get-<Noun> -Filter 'VolumeLevel -like "[123]"'
Get-<Noun> -Filter 'Enabled -ne $false'
Get-<Noun> -Filter 'NullableProperty -ne $null'
```

You can check boolean values without an explicit comparison operator, and you can also combine them with `-not`:

```
Get-<Noun> -Filter 'Enabled' # Equivalent to 'Enabled -eq $true'
Get-<Noun> -Filter '-not Enabled' # Equivalent to 'Enabled -eq $false'
```

See [about\\_Comparison\\_Operators](#) for an explanation of the operators, but note that only a subset of PowerShell operators are supported (`-eq`, `-ne`, `-gt`, `-ge`, `-lt`, `-le`, `-like`, `-notlike`, `-in`, `-notin`, `-contains`, `-notcontains`).

Enumeration values can either be specified using typed values or the string name of the enumeration value:

```
Get-<Noun> -Shape [Shapes]::Square
Get-<Noun> -Shape Circle
```

With filter expressions, typed values can be specified with simple variables or quoted strings. They also support enumerations with wildcards:

```
$s = [Shapes]::Square
Get-<Noun> -Filter { Shape -eq $s -or Shape -eq "Circle" }
Get-<Noun> -Filter { Shape -like 'C*' }
```

By their nature, floating point values, DateTime values, and TimeSpan values are best suited to relative comparisons rather than just equality. DateTime strings are converted using the locale and time zone of the user device, but you can use ISO8601 format strings (YYYY-MM-DDThh:mm:ss.sTZD) to avoid ambiguity. You can also use standard PowerShell syntax to create these values:

```
Get-<Noun> -Filter { StartTime -ge "2010-08-23T12:30:00.0Z" }
$d = [DateTime]"2010-08-23T12:30:00.0Z"
Get-<Noun> -Filter { StartTime -ge $d }
$d = (Get-Date).AddDays(-1)
Get-<Noun> -Filter { StartTime -ge $d }
```

Relative times are quite common and, when using filter expressions, you can also specify DateTime values using a relative format:

```
Get-<Noun> -Filter { StartTime -ge '-2' } # Two days ago
Get-<Noun> -Filter { StartTime -ge '-1:30' } # Hour and a half ago
Get-<Noun> -Filter { StartTime -ge '-0:0:30' } # 30 seconds ago
```

## ARRAY PROPERTIES

When filtering against list or array properties, simple parameters perform a case-insensitive wildcard match against each of the members. With filter expressions, you can use the -contains and -notcontains operators. Unlike PowerShell, these perform wildcard matching on strings.

Note that for array properties the naming convention is for the returned property to be plural, but the parameter used to search for any match is singular. The following are equivalent (assuming Users is an array property):

```
Get-<Noun> -User Fred*
Get-<Noun> -Filter { User -like "Fred*" }
Get-<Noun> -Filter { Users -contains "Fred*" }
```

You can also use the singular form with -Filter to search using other operators:

```
# Match if any user in the list is called "Frederick"
Get-<Noun> -Filter { User -eq "Frederick" }
# Match if any user in the list has a name alphabetically below 'F'
Get-<Noun> -Filter { User -lt 'F' }
```

## COMPLEX EXPRESSIONS

When matching against multiple values, you can use a sequence of comparisons joined with -or operators, or you can use -in and -notin:

```
Get-<Noun> -Filter { Shape -eq 'Circle' -or Shape -eq 'Square' }
$shapes = 'Circle','Square'
Get-<Noun> -Filter { Shape -in $shapes }
$sides = 1..4
Get-<Noun> -Filter { Sides -notin $sides }
```

Braces can be used to group complex expressions, and override the default left-to-right evaluation of -and and -or. You can also use -not to invert the sense of any sub-expression:

```
Get-<Noun> -Filter { Size -gt 4 -or (Color -eq 'Blue' -and Shape -eq 'Circle') }
Get-<Noun> -Filter { Sides -lt 5 -and -not (Color -eq 'Blue' -and Shape -eq 'Circle') }
```

## PAGING

The simplest way to page through data is to use the -Skip and -MaxRecordCount parameters. So, to read the first three pages of data with 10 records per page, use:

```
Get-<Noun> -Skip 0 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 10 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 20 -MaxRecordCount 10 <other filtering criteria>
```

You must include the same filtering criteria on each call, and ensure that the data is sorted consistently.

The above approach is often acceptable, but as each call performs an independent query, data changes can result in records being skipped or appearing twice. One approach to improve this is to sort by a unique id field and then start the search for the next page at the unique id after the last unique id of the previous page. For example:

```
# Get the first page
Get-<Noun> -MaxRecordCount 10 -SortBy SerialNumber

SerialNumber ...
----- ---
A120004
A120007
... 7 other records ...
A120900

# Get the next page
Get-<Noun> -MaxRecordCount 10 -Filter { FirstName -gt 'A120900' }

SerialNumber ...
```

----- ---  
A120901  
B220000  
...

## FILTER SYNTAX DEFINITION

<Filter> ::= <ScriptBlock> | <ComponentList>

<ScriptBlock> ::= "{" <ComponentList> "}"

<ComponentList> ::= <Component> <AndOrOperator> <ComponentList> |

<Component>

<Component> ::= <NotOperator> <Factor> |

<Factor>

<Factor> ::= "(" <ComponentList> ")" |

<PropertyName> <ComparisonOperator> <Value> |  
<PropertyName>

<AndOrOperator> ::= "-and" | "-or"

<NotOperator> ::= "-not" | "!"

<ComparisonOperator>

::= "-eq" | "-ne" | "-le" | "-ge" | "-lt" | "-gt" |  
"-like" | "-notlike" | "-contains" | "-notcontains" |  
"-in" | "-notin"

<PropertyName> ::= <simple name of property>

<Value> ::= <string literal> | <numeric literal> |

<scalar variable> | <array variable> |  
"\$null" | "\$true" | "\$false"



Numeric literals support decimal and hexadecimal literals, with optional multiplier suffixes (kb, mb, gb, tb, pb).

Dates and times can be specified as string literals. The current culture determines what formats are accepted. To avoid any ambiguity, use strings formatted to the ISO8601 standard. If not specified, the current time zone is used.

Relative date-time string literals are also supported, using a minus sign followed by a TimeSpan. For example, "-1:30" means 1 hour and 30 minutes ago.

# Add-AcctADAccount

Jan 04, 2017

Import Active Directory computer accounts from Active Directory for use in the AD Identity Service.

## Syntax

```
Add-AcctADAccount [-IdentityPoolName] <String> -ADAccountName <String[]> [-Password <String>] [-SecurePassword <SecureString>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-AcctADAccount -IdentityPoolUid <Guid> -ADAccountName <String[]> [-Password <String>] [-SecurePassword <SecureString>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for Active Directory computer accounts that already exist in Active Directory to be used by the Citrix AD Identity Service and the other Citrix Machine Creation Services.

All aspects of this command that need to make modifications to the accounts in Active Directory will do so using the account that the PowerShell runspace is using. This means that if the passwords need resetting for the accounts, the user performing the operation in PowerShell must have sufficient privileges in Active Directory for this operation to complete successfully.

The following rules apply to the importing of Active Directory accounts; If the current account password is supplied, the cmdlet will attempt to change the password so that it is known only to the Citrix Identity Service. This uses password change operations and does not need AD account administration permissions. If the current password is not supplied, the cmdlet will attempt to reset the password for the Active Directory account so that it is known only to the Citrix Identity Service. This requires the cmdlet to have enough privileges in Active Directory for the accounts' password reset to be available. Imported accounts in a disabled or locked state in Active Directory are imported with the account marked in an error state. If the identity pool into which the account is being imported does not have a domain set, it assumes the domain of the first account imported into it.

## Related topics

[New-AcctADAccount](#)

[Remove-AcctADAccount](#)

[Repair-AcctADAccount](#)

[Get-AcctADAccount](#)

[Update-AcctADAccount](#)

[Unlock-AcctADAccount](#)

## Parameters

**-IdentityPoolName**<String>

The identity pool name into which to add the imported accounts.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-ADAccountName**<String[]>

The Active Directory account name to be imported.

Active Directory accounts are accepted in the following formats: Fully qualified DN e.g.

CN=MyComputer,OU=Computers,DC=MyDomain,DC=Com; UPN format e.g MyComputer@MyDomain.Com; Domain qualified e.g MyDomain\MyComputer.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IdentityPoolUid**<Guid>

The unique identifier for the identity pool to which imported accounts are to be added.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Password**<String>

The current password for the computer account.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecurePassword**<SecureString>

The current password for the account (provided in a Secure String class).

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | LocalHost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Return Values

Citrix.AdIdentity.Sdk.AccountOperationDetailedSummary

The Add-AcctADAccount returns an object that contains the following parameters;

SuccessfulAccountsCount <int>

The number of accounts that were added successfully

FailedAccountsCount <int>

The number of accounts that were not added.

FailedAccounts <Citrix.XDPowerShell.AccountError[]>

The list of accounts that failed to be added. Each one has the following properties;

ADAccountName <string>

ADAccountSid <String>

ErrorReason <AdIdentityStatus>

This can be one of the following

UnableToConvertDomain

UnableToAccessAccountProperties

IdentityNotLocatedInDomain

UnableToAccessAccountProperties

IdentityDuplicateObjectExists

IdentityObjectLocked

IdentityObjectInUse

FailedToConnectToDomainController

FailedToExecuteSearchInAD

FailedToAccessComputerAccountInAD

FailedToSetPasswordInAD

FailedToChangePasswordInAD

ADServiceDatabaseError

ADServiceDatabaseNotConfigured

ADServiceStatusInvalidDb

DiagnosticInformation <Exception>

Any other error information

SuccessfulAccounts <Citrix.AdIdentity.Sdk.Identity[]>

The list of accounts that were successfully added. Each one has the following properties;

ADAccountSid <string>

The AD account SID for the imported account.

ADAccountName <string>

The AD account name for the imported account.

Domain <string>

The domain for the imported account.

State <Citrix.AdIdentity.Sdk.ADIdentityState>

The state for the account. This can be;

Available

The account is not used.

InUse

The account is in use.

Error

The account is in error (i.e. the account is locked or disabled in AD).

Tainted

The account is no longer used, but the password is no longer known.

Lock <Boolean>

The account is locked (in the database not in AD).

Notes

To maintain maximum security when using the command programmatically, Citrix recommends you use the 'SecurePassword' property instead of the 'Password' property.

In the case of failure, the following errors can result.

Error Codes

-----

IdentityPoolAlreadyLocked

The specified identity pool was locked by another operation.

IdentityPoolNotFound

The specified identity pool was not found.

IdentityDuplicateObjectExists

The specified AD account already exists for the identity pool.

PermissionDenied

The user does not have administrative rights to perform this operation.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

CommunicationError

An error occurred while communicating with the service.

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used, or examine the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
C:\PS>Add-AcctADAccount -IdentityPoolName MyPool -ADAccountName "Domain\account","Domain\account2" -OutVariable result
```

| SuccessfulAccounts | SuccessfulAccountsCount | FailedAccountsCount | FailedAccounts |
|--------------------|-------------------------|---------------------|----------------|
|--------------------|-------------------------|---------------------|----------------|

|                                   |   |   |    |
|-----------------------------------|---|---|----|
| {domain\account, domain\account2} | 2 | 0 | {} |
|-----------------------------------|---|---|----|

```
$result[0].SuccessfulAccounts
```

```
ADAccountSid : S-1-5-21-1315084875-1285793635-2418178940-2644
```

```
ADAccountName : domain\account
```

```
Domain       : Domain.com
```

```
State        : Available
```

```
Lock         : False
```

```
ADAccountSid : S-1-5-21-1315084875-1285793635-2418178940-2645
```

```
ADAccountName : domain\account2
```

```
Domain       : Domain.com
```

```
State        : Available
```

```
Lock         : False
```

Import the two accounts (account and account2) from AD into the identity Pool called "MyPool"

# Add-AcctIdentityPoolScope

Jan 04, 2017

Add the specified IdentityPool(s) to the given scope(s).

## Syntax

```
Add-AcctIdentityPoolScope [-Scope] <String[]> -InputObject <IdentityPool[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-AcctIdentityPoolScope [-Scope] <String[]> -IdentityPoolUid <Guid[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-AcctIdentityPoolScope [-Scope] <String[]> -IdentityPoolName <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The AddAcctIdentityPoolScope cmdlet is used to associate one or more IdentityPool objects with given scope(s).

There are multiple parameter sets for this cmdlet, allowing you to identify the IdentityPool objects in different ways:

- IdentityPool objects can be piped in or specified by the InputObject parameter
- The IdentityPoolUid parameter specifies objects by IdentityPoolUid
- The IdentityPoolName parameter specifies objects by IdentityPoolName (supports wildcards)

To add a IdentityPool to a scope you need permission to change the scopes of the IdentityPool and permission to add objects to all of the scopes you have specified.

If the IdentityPool is already in a scope, that scope will be silently ignored.

## Related topics

[Remove-AcctIdentityPoolScope](#)

[Get-AcctScopedObject](#)

## Parameters

**-Scope**<String[]>

Specifies the scopes to add the objects to.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-InputObject**<IdentityPool[]>

Specifies the IdentityPool objects to be added.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-IdentityPoolUid**<Guid[]>

Specifies the IdentityPool objects to be added by IdentityPoolUid.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-IdentityPoolName**<String[]>

Specifies the IdentityPool objects to be added by IdentityPoolName.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

None

### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

##### UnknownObject

One of the specified objects was not found.

##### ScopeNotFound

One of the specified scopes was not found.

##### DatabaseError

An error occurred in the service while attempting a database operation.

##### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

##### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

##### PermissionDenied

You do not have permission to execute this command with the specified objects or scopes.

##### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Add-AcctIdentityPoolScope Finance -IdentityPoolUid 6702C5D0-C073-4080-A0EE-EC74CB537C52
```

Adds a single IdentityPool to the 'Finance' scope.

### ----- EXAMPLE 2 -----

```
c:\PS>Add-AcctIdentityPoolScope Finance,Marketing -IdentityPoolUid 6702C5D0-C073-4080-A0EE-EC74CB537C52
```

Adds a single IdentityPool to the multiple scopes.

### ----- EXAMPLE 3 -----

```
c:\PS>Get-AcctIdentityPool | Add-AcctIdentityPoolScope Finance
```

Adds all visible IdentityPool objects to the 'Finance' scope.

### ----- EXAMPLE 4 -----

```
c:\PS>Add-AcctIdentityPoolScope Finance -IdentityPoolName A*
```

Adds IdentityPool objects with a name starting with an 'A' to the 'Finance' scope.

# Copy-AcctIdentityPool

Jan 04, 2017

Copies an Identity Pool and its associated Identities to a new IdentityPool

## Syntax

```
Copy-AcctIdentityPool [-IdentityPoolName] <String> [-NewIdentityPoolName] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Copy-AcctIdentityPool -IdentityPoolUid <Guid> [-NewIdentityPoolName] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to copy an IdentityPool.

The new IdentityPool will contain all the accounts that were in the original pool and will have the same domain and OU set. The naming scheme will be unset and the StartCount will be set to 1.

## Related topics

[New-AcctIdentityPool](#)

[Get-AcctIdentityPool](#)

[Remove-AcctIdentityPool](#)

## Parameters

**-IdentityPoolName**<String>

The name of the identity pool that is to be copied.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IdentityPoolUid**<Guid>

The unique identifier for the identity pool that is to be copied.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-NewIdentityPoolName**<String>

The name for the new IdentityPool.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

Citrix.AdIdentity.Sdk.IdentityPool

This object provides details of the new identity pool and contains the following information:

IdentityPoolName <string>

The name of the identity pool.

IdentityPoolUid <Guid>

The unique identifier for the identity pool.

NamingScheme <string>

The naming scheme for the identity pool.

NamingSchemeType <Citrix.XDIInterServiceTypes.ADIIdentityNamingScheme>

The naming scheme type for the identity pool. This can be one of the following:

Numeric - naming scheme uses numeric indexes

Alphabetic - naming scheme uses alphabetic indexes

StartCount <int>

The next index to be used when creating an identity from the identity pool.

OU <string>

The Active Directory distinguished name for the OU in which accounts created from this identity pool will be created.

Domain <string>

The Active Directory domain that accounts in the pool belong to.

Lock <Boolean>

Indicates whether the identity pool is locked.

Notes

In the case of failure, the following errors can result.

Error Codes

-----

IdentityPoolDuplicateObjectExists

An identity pool with the same name exists already.

IdentityPoolObjectNotFound

The identity pool to be modified could not be located.

PermissionDenied

The user does not have administrative rights to perform this operation.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

# Get-AcctADAccount

Jan 04, 2017

Gets the AD accounts stored in the AD Identity Service.

## Syntax

```
Get-AcctADAccount [-IdentityPoolName <String>] [-ADAccountSid <String>] [-Domain <String>] [-State <ADIdentityState>] [-Lock <Boolean>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-AcctADAccount [-IdentityPoolUid <Guid>] [-ADAccountSid <String>] [-Domain <String>] [-State <ADIdentityState>] [-Lock <Boolean>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to locate the AD accounts stored within the AD Identity Service and view the state of the accounts.

## Related topics

[New-AcctADAccount](#)

[Add-AcctADAccount](#)

[Remove-AcctADAccount](#)

[Unlock-AcctADAccount](#)

[Update-AcctADAccount](#)

[Repair-AcctADAccount](#)

## Parameters

**-ADAccountSid**<String>

The AD Account SID of the account.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Domain**<String>

The domain of the account (this is in dns format).

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-State**<ADIdentityState>

The current state of the identity stored in the AD Identity Service for the AD account.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Lock**<Boolean>

Indicates if the account is locked in the AD Identity Service.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

See about\_Acct\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

See about\_Acct\_Filtering for details.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

See about\_Acct\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

See about\_Acct\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Filter**<String>

See about\_Acct\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | LocalHost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-IdentityPoolName**<String>

The name of the identity pool to which the account is registered.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IdentityPoolUid**<Guid>

The unique identifier for the identity pool that the account is registered to.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

Return Values

Citrix.AdIdentity.Sdk.IdentityInPool

The Get-AcctADAccount returns an object that contains the following parameters

ADAccountSid <string>

The AD account SID for the retrieved account.

ADAccountName <string>

The AD account name for the retrieved account.

Domain <string>

The domain for the imported account.

State <Citrix.XDInterServiceTypes.ADIdentityState>

The state for the account. This can be;

Available

The account is not used.

## InUse

The account is in use.

## Error

The account is in error (i.e. the account is locked or disabled in AD).

## Tainted

The account is no longer used, but the password is no longer known.

## Lock <Boolean>

The account is locked (in the database not in AD).

## IdentityPoolName <System.String>

The name of the containing identity pool.

## IdentityPoolUid <System.Guid>

The GUID identifying the containing identity pool.

## Notes

In the case of failure the following errors can result.

## Error Codes

-----

### PartialData

Only a subset of the available data was returned.

### CouldNotQueryDatabase

The query required to get the database was not defined.

### PermissionDenied

The user does not have administrative rights to perform this operation.

### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

### CommunicationError

An error occurred while communicating with the service.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

## ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\>Get-AcctADAccount
```

Return all the AD accounts that are registered in the AD Identity Service.

### ----- EXAMPLE 2 -----

```
C:\>Get-AcctADAccount -IdentityPoolName MyPool -Lock $false
```

Return all the AD accounts that are registered in the AD Identity Service in the identity pool called "MyPool" that are also locked.

### ----- EXAMPLE 3 -----

```
C:\>Get-AcctADAccount -Filter {IdentityPoolName -Like "p*" -or IdentityPoolName -eq "MyPool"}
```

Return all the AD accounts that are registered in the AD Identity Service in the identity pool called "MyPool" or in an identity pool that has a name that starts with a 'p'. For full details of the advanced filtering aspects of this command see about\_Acct\_Filtering.

# Get-AcctDBConnection

Jan 04, 2017

Gets the database string for the specified data store used by the AdIdentity Service.

## Syntax

```
Get-AcctDBConnection [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the database connection string for the specified data store.

If the returned string is blank, no valid connection string has been specified. In this case the service is running, but is idle and awaiting specification of a valid connection string.

## Related topics

[Get-AcctServiceStatus](#)

[Set-AcctDBConnection](#)

[Test-AcctDBConnection](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

system.string

The database connection string configured for the AdIdentity Service.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

NoDBConnections

The database connection string for the AdIdentity Service has not been specified.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AcctDBConnection
```

Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True  
Get the database connection string for the AdIdentity Service.

# Get-AcctDBSchema

Jan 04, 2017

Gets a script that creates the AdIdentity Service database schema for the specified data store.

## Syntax

```
Get-AcctDBSchema [-DatabaseName <String>] [-ServiceGroupName <String>] [-ScriptType <ScriptTypes>] [-LocalDatabase] [-Sid <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets SQL scripts that can be used to create a new AdIdentity Service database schema, add a new AdIdentity Service to an existing site, remove a AdIdentity Service from a site, or create a database server logon for a AdIdentity Service. If no Sid parameter is provided, the scripts obtained relate to the currently selected AdIdentity Service instance, otherwise the scripts relate to AdIdentity Service instance running on the machine identified by the Sid provided. When obtaining the Evict script, a Sid parameter must be supplied. The current service instance is that on the local machine, or that explicitly specified by the last usage of the -AdminAddress parameter to a AdIdentity SDK cmdlet. The service instance used to obtain the scripts does not need to be a member of a site or to have had its database connection configured. The database scripts support only Microsoft SQL Server, or SQL Server Express, and require Windows integrated authentication to be used. They can be run using SQL Server's SQLCMD utility, or by copying the script into an SQL Server Management Studio (SSMS) query window and executing the query. If using SSMS, the query must be executed in 'SMDCMD mode'. The ScriptType parameter determines which script is obtained. If ScriptType is not specified, or is FullDatabase, the script contains:

- o Creation of service schema
- o Creation of database server logon
- o Creation of database user
- o Addition of database user to AdIdentity Service roles

If ScriptType is Instance, the returned script contains:

- o Creation of database server logon
- o Creation of database user
- o Addition of database user to AdIdentity Service roles

If ScriptType is Evict, the returned script contains:

- o Removal of AdIdentity Service instance from database
- o Removal of database user

If ScriptType is Login, the returned script contains:

- o Creation of database server logon only

If the service uses two data stores they can exist in the same database. You do not need to configure a database before using this command.



## Related topics

[Set-AcctDBConnection](#)

## Parameters

### **-DatabaseName**<String>

Specifies the name of the database for which the schema will be generated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-ServiceGroupName**<String>

Specifies the name of the service group to be used when creating the database schema. The service group is a collection of all the AdIdentity services that share the same database instance and are considered equivalent; that is, all the services within a service group can be used interchangeably.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-Script Type**<ScriptTypes>

Specifies the type of database script returned. Available script types are:

#### Database

Returns a full database script that can be used to create a database schema for the AdIdentity Service in a database instance that does not already contain a schema for this service. The DatabaseName and ServiceGroupName parameters must be specified to create a script of this type.

#### Instance

Returns a permissions script that can be used to add further AdIdentity services to an existing database instance that already contains the full AdIdentity service schema, associating the services to the Service Group. The Sid parameter can optionally be specified to create a script of this type.

#### Login

Returns a database logon script that can be used to add the required logon accounts to an existing database instance that contains the AdIdentity Service schema. This is used primarily when creating a mirrored database environment. The

DatabaseName parameter must be specified to create a script of this type.

#### Evict

Returns a script that can be used to remove the specified AdIdentity Service from the database entirely. The DatabaseName and Sid parameters must be specified to create a script of this type.

|                        |          |
|------------------------|----------|
| Required?              | false    |
| Default Value          | Database |
| Accept Pipeline Input? | false    |

#### -LocalDatabase<SwitchParameter>

Specifies whether the database script is to be used in a database instance run on the same controller as other services in the service group. Including this parameter ensures the script creates only the required permissions for local services to access the database schema for AdIdentity services.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### -Sid<String>

Specifies the SID of the controller on which the AdIdentity Service instance to remove from the database is running.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

#### -AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

### Systemstring

A string containing the required SQL script for application to a database.

## Notes

The scripts returned support Microsoft SQL Server Express Edition, Microsoft SQL Server Standard Edition, and Microsoft SQL Server Enterprise Edition databases only, and are generated on the assumption that integrated authentication will be used.

If the ScriptType parameter is not included or set to 'FullDatabase', the full database script is returned, which will:

Create the database schema.

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist).

If the ScriptType parameter is set to 'Instance', the script will:

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist) and associate it with a user.

If the ScriptType parameter is set to 'Login', the script will:

Create the logon (providing the schema does not already exist) and associate it with a pre-existing user of the same name.

If the LocalDatabase parameter is included, the NetworkService account will be added to the list of accounts permitted to access the database. This is required only if the database is run on a controller.

If the command fails, the following errors can be returned.

## Error Codes

-----

### GetSchemasFailed

The database schema could not be found.

### ActiveDirectoryAccountResolutionFailed

The specified Active Directory account or Group could not be found.

### DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-AcctDBSchema -DatabaseName MyDB -ServiceGroupName MyServiceGroup > c:\AcctSchema.sql
```

Get the full database schema for site data store of the AdIdentity Service and copy it to a file called 'c:\AcctSchema.sql'.

This script can then be used to create the schema in a pre-existing database named 'MyDB' that does not already contain a AdIdentity Service site schema.

### ----- EXAMPLE 2 -----

```
c:\PS>Get-AcctDBSchema -DatabaseName MyDB -scriptType Login > c:\AdIdentityLogins.sql
```

Get the logon scripts for the AdIdentity Service.

# Get-AcctDBVersionChangeScript

Jan 04, 2017

Gets a script that updates the AdIdentity Service database schema.

## Syntax

```
Get-AcctDBVersionChangeScript -DatabaseName <String> -TargetVersion <Version> [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Returns a database script that can be used to upgrade or downgrade the site or secondary schema for the AdIdentity Service from the current schema version to a different version.

## Related topics

[Get-AcctInstalledDBVersion](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database instance to which the update applies.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TargetVersion**<Version>

Specifies the version of the database you want to update to.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.Management.Automation.PSObject

A PSObject containing the required SQL script for application to a database.

## Notes

The PSObject returned by this cmdlet contains the following properties:

- Script The raw text of the SQL script to apply the update, or null in the case when no upgrade path to the specified target version exists.
- NeedExclusiveAccess Indicates whether all services in the service group must be shut down during the update or not.
- CanUndo Indicates whether the generated script allows the updated schema to be reverted to the state prior to the update.

Scripts to update the schema version are stored in the database so any service in the service group can obtain these scripts. Extreme caution should be exercised when using update scripts. Citrix recommends backing up the database before attempting to upgrade the schema. Database update scripts may require exclusive use of the schema and so may not be able to execute while any AdIdentity services are running. However, this depends on the specific update being carried out.

After a schema update has been carried out, services that require the previous version of the schema may cease to operate. The ServiceState parameter reported by the Get-AcctServiceStatus command provides information about service compatibility. For example, if the schema has been upgraded to a more recent version that a service cannot use, the service reports "DBNewerVersionThanService".

If the command fails, the following errors can be returned.

## Error Codes

-----

### NoOp

The operation was successful but had no effect.

### NoDBConnections

The database connection string for the AdIdentity Service has not been specified.

### DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $update = Get-AcctDBVersionChangeScript -DatabaseName MyDb -TargetVersion 1.0.75.0
```

```
C:\PS> $update.Script > update_75.sql
```

Gets an SQL update script to update the current schema to version 1.0.75.0. The resulting update\_75.sql script is suitable for direct use with the SQL Server SQLCMD utility.

# Get-AcctIdentityPool

Jan 04, 2017

Gets existing identity pools.

## Syntax

```
Get-AcctIdentityPool [-IdentityPoolName <String>] [-IdentityPoolUid <Guid>] [-Lock <Boolean>] [-Scopeld <Guid>] [-ScopeName <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to locate existing identity pools.

## Related topics

[New-AcctIdentityPool](#)

[Remove-AcctIdentityPool](#)

[Rename-AcctIdentityPool](#)

[Set-AcctIdentityPool](#)

## Parameters

**-IdentityPoolName**<String>

The name of the identity pool.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IdentityPoolUid**<Guid>

The unique identifier for the identity pool.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-Lock**<Boolean>

Whether the identity pool is locked or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ScopeId**<Guid>

Gets only results with a scope matching the specified scope identifier.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ScopeName**<String>

Gets only results with a scope matching the specified scope name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

See about\_Acct\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

See about\_Acct\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

See about\_Acct\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

See about\_Acct\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Filter**<String>

See about\_Acct\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host

name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | LocalHost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Return Values

Citrix.AdIdentity.Sdk.IdentityPool

This object provides details of the identity pool and contains the following information:

IdentityPoolName <string>

The name of the identity pool.

IdentityPoolUid <Guid>

The unique identifier for the identity pool.

NamingScheme <string>

The naming scheme for the identity pool.

NamingSchemeType <Citrix.XDInterServiceTypes.ADIdentityNamingScheme>

The naming scheme type for the identity pool. This can be one of the following:

Numeric - naming scheme uses numeric indexes

Alphabetic - naming scheme uses alphabetic indexes

StartCount <int>

The next index to be used when creating an identity from the identity pool.

OU <string>

The Active Directory distinguished name for the OU in which accounts created from this identity pool will be created.

Domain <string>

The Active Directory domain that accounts in the pool belong to.

Lock <Boolean>

Indicates if the identity pool is locked.

## Notes

In the case of failure, the following errors can result.

## Error Codes

-----

### PartialData

Only a subset of the available data was returned.

### CouldNotQueryDatabase

The query required to get the database was not defined.

### PermissionDenied

The user does not have administrative rights to perform this operation.

### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

### CommunicationError

An error occurred while communicating with the service.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS>Get-AcctIdentityPool
```

```
IdentityPoolName : MyPool
IdentityPoolUid  : 22072d9e-6a8f-494b-a5bc-2ef18ca4b915
NamingScheme    : Acc####
NamingSchemeType : Numeric
StartCount      : 1
OU              :
Domain          : mydomain.com
Lock            : True
```

```
IdentityPoolName : MyPool2
IdentityPoolUid  : 03743136-e43b-4a87-af74-ab71686b3c16
```

NamingScheme : Test####  
NamingSchemeType : Alphabetic  
StartCount : 1  
OU :  
Domain : mydomain.com  
Lock : False  
Gets all the identity pools.

----- **EXAMPLE 2** -----

C:\PS>Get-AcctIdentityPool -IdentityPoolName M\*

IdentityPoolName : MyPool  
IdentityPoolUid : 22072d9e-6a8f-494b-a5bc-2ef18ca4b915  
NamingScheme : Acc####  
NamingSchemeType : Numeric  
StartCount : 1  
OU :  
Domain : mydomain.com  
Lock : True  
Gets all the identity pools beginning with the character 'M'.

# Get-AcctInstalledDBVersion

Jan 04, 2017

Gets a list of all available database schema versions for the AdIdentity Service.

## Syntax

```
Get-AcctInstalledDBVersion [-Upgrade] [-Downgrade] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the current version of the AdIdentity Service database schema, if no flags are set, otherwise returns versions for which upgrade or downgrade scripts are available and have been stored in the database.

## Related topics

## Parameters

**-Upgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be updated should be returned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Downgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be reverted should be returned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

### System.Version

The Get-AcctInstalledDbVersion command returns objects containing the new definition of the AdIdentity Service database schema version.

Major <Integer>

Minor <Integer>

Build <Integer>

Revision <Integer>

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

Both the Upgrade and Downgrade flags were specified.

#### NoOp

The operation was successful but had no effect.

#### NoDBConnections

The database connection string for the AdIdentity Service has not been specified.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-AcctInstalledDBVersion
```

```
Major Minor Build Revision
```

```
-----
```

```
5 6 0 0
```

Get the currently installed version of the AdIdentity Service database schema.

### ----- EXAMPLE 2 -----

```
c:\PS>Get-AcctInstalledDBVersion -Upgrade
```

```
Major Minor Build Revision
```

```
-----
```

```
6 0 0 0
```

Get the versions of the AdIdentity Service database schema for which upgrade scripts are supplied.



# Get-AcctScopedObject

Jan 04, 2017

Gets the details of the scoped objects for the AdIdentity Service.

## Syntax

```
Get-AcctScopedObject [-ScopeId <Guid>] [-ScopeName <String>] [-ObjectType <ScopedObjectType>] [-ObjectId <String>] [-ObjectName <String>] [-Description <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns a list of directly scoped objects including the names and identifiers of both the scope and object as well as the object description for display purposes.

There will be at least one result for every directly scoped object. When an object is associated with multiple scopes the output contains one result per scope duplicating the object details.

No records are returned for the All scope, though if an object is not in any scope a result with a null ScopeId and ScopeName will be returned.

## Related topics

### Parameters

#### **-ScopeId**<Guid>

Gets scoped object entries for the given scope identifier.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

#### **-ScopeName**<String>

Gets scoped object entries with the given scope name.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ObjectType**<ScopedObjectType>

Gets scoped object entries for objects of the given type.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ObjectId**<String>

Gets scoped object entries for objects with the specified object identifier.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ObjectName**<String>

Gets scoped object entries for objects with the specified object identifier.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Description**<String>

Gets scoped object entries for objects with the specified description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Acct\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

### **-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

### **-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Acct\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

Citrix.AdIdentity.Sdk.ScopedObject

The Get-AcctScopedObject command returns an object containing the following properties:

ScopeId <Guid?>

Specifies the unique identifier of the scope.

ScopeName <String>

Specifies the display name of the scope.

ObjectType <ScopedObjectType>

Type of the object this entry relates to.

ObjectId <String>

Unique identifier of the object.

ObjectName <String>

Display name of the object

Description <String>

Description of the object (possibly \$null if the object type does not have a description).

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

UnknownObject

One of the specified objects was not found.

PartialData

Only a subset of the available data was returned.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

CouldNotQueryDatabase

The query required to get the database was not defined.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AcctScopedObject -ObjectType Scheme
```

```
ScopelId   : eff6f464-f1ee-4442-add3-99982e0cec01
ScopeName  : Sales
ObjectType : Scheme
ObjectId   : cd4174ee-9e4b-4e57-b126-9dbf757fe493
ObjectName : MyExampleScheme
Description : Test scheme
```

```
ScopelId   : 304e0fa7-d390-47f0-a94f-7e956a324c41
ScopeName  : Finance
ObjectType : Scheme
ObjectId   : cd4174ee-9e4b-4e57-b126-9dbf757fe493
ObjectName : MyExampleScheme
Description : Test scheme
```

```
ScopelId   :
ScopeName  :
ObjectType : Scheme
ObjectId   : 5062e46b-71bc-4ac9-901a-30fe6797e2f6
ObjectName : AnotherScheme
Description : Another scheme in no scopes
```

Gets all of the scoped objects with type Scheme. The example output shows a scheme object (MyExampleScheme) in two scopes Sales and Finance, and another scheme (AnotherScheme) that is not in any scope. The ScopelId and ScopeName values returned are null in the final record.

# Get-AcctService

Jan 04, 2017

Gets the service record entries for the AdIdentity Service.

## Syntax

```
Get-AcctService [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns instances of the AdIdentity Service that the service publishes. The service records contain account security identifier information that can be used to remove each service from the database.

A database connection for the service is required to use this command.

## Related topics

### Parameters

#### **-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Acct\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-Sort By**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |



### **-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Acct\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

Citrix.AdIdentity.Sdk.Service

The Get-AcctServiceInstance command returns an object containing the following properties.

Uid <Integer>

Specifies the unique identifier for the service in the group. The unique identifier is an index number.

ServiceHostId <Guid>

Specifies the unique identifier for the service instance.

DNSName <String>

Specifies the domain name of the host on which the service runs.

MachineName <String>

Specifies the short name of the host on which the service runs.

CurrentState <Citrix.Fma.Sdk.ServiceCore.ServiceState>

Specifies whether the service is running, started but inactive, stopped, or failed.

LastStartTime <DateTime>

Specifies the date and time at which the service was last restarted.

LastActivityTime <DateTime>

Specifies the date and time at which the service was last stopped or restarted.

OSType

Specifies the operating system installed on the host on which the service runs.

OSVersion

Specifies the version of the operating system installed on the host on which the service runs.

ServiceVersion

Specifies the version number of the service instance. The version number is a string that reflects the full build version of the service.

DatabaseUserName <string>

Specifies for the service instance the Active Directory account name with permissions to access the database. This will be either the machine account or, if the database is running on a controller, the NetworkService account.

Sid <string>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

ActiveSiteServices <string[]>

Specifies the names of active site services currently running in the service. Site services are components that perform long-running background processing in some services. This field is empty for services that do not contain site services.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### PartialData

Only a subset of the available data was returned.

#### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

#### CouldNotQueryDatabase

The query required to get the database was not defined.

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-AcctService
```

```
Uid          : 1
ServiceHostId : aef6f464-f1ee-4042-a523-66982e0cecd0
DNSName      : MyServer.company.com
MachineName  : MYSERVER
CurrentState  : On
LastStartTime : 04/04/2011 15:25:38
LastActivityTime : 04/04/2011 15:33:39
OSType       : Win32NT
OSVersion    : 6.1.7600.0
ServiceVersion : 5.1.0.0
DatabaseUserName : NT AUTHORITY\NETWORK SERVICE
SID          : S-1-5-21-2316621082-1546847349-2782505528-1165
ActiveSiteServices : {MySiteService1, MySiteService2...}
Get all the instances of the AdIdentity Service running in the current service group.
```

# Get-AcctServiceAddedCapability

Jan 04, 2017

Gets any added capabilities for the AdIdentity Service on the controller.

## Syntax

```
Get-AcctServiceAddedCapability [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables updates to the AdIdentity Service on the controller to be detected.

You do not need to configure a database connection before using this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.String

String containing added capabilities.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-AcctServiceAddedCapability
```

Get the added capabilities of the AdIdentity Service.

# Get-AcctServiceInstance

Jan 04, 2017

Gets the service instance entries for the AdIdentity Service.

## Syntax

```
Get-AcctServiceInstance [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns service interfaces published by the instance of the AdIdentity Service. Each instance of a service publishes multiple interfaces with distinct interface types, and each of these interfaces is represented as a ServiceInstance object. Service instances can be used to register the service with a central configuration service so that other services can use the functionality.

You do not need to configure a database connection to use this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.AdIdentity.Sdk.ServiceInstance

The Get-AcctServiceInstance command returns an object containing the following properties.

ServiceGroupUid <Guid>

Specifies the unique identifier for the service group of which the service is a member.

ServiceGroupName <String>

Specifies the name of the service group of which the service is a member.

ServiceInstanceUID <Guid>

Specifies the unique identifier for registered service instances, which are service instances held by and obtained from a

central configuration service. Unregistered service instances do not have unique identifiers.

ServiceType <String>

Specifies the service instance type. For this service, the service instance type is always Acct.

Address

Specifies the address of the service instance. The address can be used to access the service and, when registered in the central configuration service, can be used by other services to access the service.

Binding

Specifies the binding type that must be used to communicate with the service instance. In this release of XenDesktop, the binding type is always 'wcf\_HTTP\_kerb'. This indicates that the service provides a Windows Communication Foundation endpoint that uses HTTP binding with integrated authentication.

Version

Specifies the version of the service instance. The version number is used to ensure that the correct versions of the services are used for communications.

ServiceAccount <String>

Specifies the Active Directory account name for the machine on which the service instance is running. The account name is used to provide information about the permissions required for interservice communications.

ServiceAccountSid <String>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

InterfaceType <String>

Specifies the interface type. Each service can provide multiple service instances, each for a different purpose, and the interface defines the purpose. Available interfaces are:

SDK - for PowerShell operations

InterService - for operations between different services

Peer - for communications between services of the same type

Metadata <Citrix.AdIdentity.Sdk.Metadata[]>

The collection of metadata associated with registered service instances, which are service instances held by and obtained from a central configuration service. Metadata is not stored for unregistered service instances.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-AcctServiceInstance
```

```
Address      : http://MyServer.com:80/Citrix/AdIdentityService
Binding      : wcf_HTTP_kerb
InterfaceType : SDK
Metadata     :
MetadataMap  :
ServiceAccount : ENG\MyAccount$
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164
ServiceGroupName : MyServiceGroup
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000
ServiceType   : Acct
Version      : 1

Address      : http://MyServer.com:80/Citrix/AdIdentityService/IServiceApi
Binding      : wcf_HTTP_kerb
InterfaceType : InterService
Metadata     :
MetadataMap  :
```



ServiceAccount : ENG\MyAccount  
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164  
ServiceGroupName : MyServiceGroup  
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d  
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000  
ServiceType : Acct  
Version : 1

Get all instances of the AdIdentity Service running on the specified machine. For remote services, use the AdminAddress parameter to define the service for which the interfaces are required. If the AdminAddress parameter has not been specified for the runspace, service instances running on the local machine are returned.

# Get-AcctServiceStatus

Jan 04, 2017

Gets the current status of the AdIdentity Service on the controller.

## Syntax

```
Get-AcctServiceStatus [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables the status of the AdIdentity Service on the controller to be monitored. If the service has multiple data stores it will return the overall state as an aggregate of all the data store states. For example, if the site data store status is OK and the secondary data store status is DBUnconfigured then it will return DBUnconfigured.

## Related topics

[Set-AcctDBConnection](#)

[Test-AcctDBConnection](#)

[Get-AcctDBConnection](#)

[Get-AcctDBSchema](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Get-AcctServiceStatus command returns an object containing the status of the AdIdentity Service together with extra diagnostics information.

DBUnconfigured

The AdIdentity Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the AdIdentity Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

#### InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the AdIdentity Service schema has not been added to the database.

#### DBNotFound

The specified database could not be located with the configured connection string.

#### DBNewerVersionThanService

The version of the AdIdentity Service currently in use is incompatible with the version of the AdIdentity Service schema on the database. Upgrade the AdIdentity Service to a more recent version.

#### DBOlderVersionThanService

The version of the AdIdentity Service schema on the database is incompatible with the version of the AdIdentity Service currently in use. Upgrade the database schema to a more recent version.

#### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

#### OK

The AdIdentity Service is running and is connected to a database containing a valid schema.

#### Failed

The AdIdentity Service has failed.

#### Unknown

(0) The service status cannot be determined.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AcctServiceStatus
```

DBUnconfigured

Get the current status of the AdIdentity Service.

# New-AcctADAccount

Jan 04, 2017

Creates AD computer accounts in the specified identity pool.

## Syntax

```
New-AcctADAccount [-IdentityPoolName] <String> -ADAccountName <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-AcctADAccount [-IdentityPoolName] <String> -Count <Int32> [-StartCount <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-AcctADAccount -IdentityPoolUid <Guid> -Count <Int32> [-StartCount <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-AcctADAccount -IdentityPoolUid <Guid> -ADAccountName <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to create new AD computer accounts and register them in an already existing identity pool.

The accounts are created using the information stored in the identity pool. This provides the account name (via the Naming Scheme property and Start Count), domain, and OU.

The runspace used for this command must have sufficient privileges in Active Directory to create the new computer accounts.

The AD account names will pad the index to use all the space specified in the identity pool naming scheme (e.g. "acc###" will become "acc001"). However, if the index overflows the available space the cmdlet expands the format to use the next incremental number (e.g. "acc###" will become "acc1000" if the index is 10000, which cannot fit into the three '#' placeholders). If this expanded name exceeds the 15 character name limit, the accounts are not created.

There can be only one creation process running for a specific identity pool at any one time. Attempting to start another account creation process while an existing one is executing results in an error being returned.

## Related topics

[Add-AcctADAccount](#)

[Remove-AcctADAccount](#)

[Get-AcctADAccount](#)

[Repair-AcctADAccount](#)

[Unlock-AcctADAccount](#)

[Update-AcctADAccount](#)

## Parameters

**-IdentityPoolName**<String>

The name of the identity pool in which to create the accounts.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ADAccount Name**<String[]>

The AD account names to be created. These are just the simple machine account names e.g. MyVM001

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Count**<Int32>

The number of accounts to create.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IdentityPoolUid**<Guid>

The unique identifier for the identity pool in which the accounts will be created.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartCount**<Int32>

The start index for the create process.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

#### Return Values

Citrix.AdIdentity.Sdk.AccountOperationDetailedSummary

The Add-AcctADAccount returns an object that contains the following parameters;

SuccessfulAccountsCount <int>

The number of accounts that were added successfully

FailedAccountsCount <int>

The number of accounts that were not added.

FailedAccounts <Citrix.AdIdentity.Sdk.AccountError[]>

The list of accounts that failed to be added. Each one has the following parameters;

ADAccountName <string>

ADAccountSid <String>

ErrorReason <string>

This can be one of the following

IdentityDuplicateObjectExists

An identity with the same SID already exists.

ADServiceDatabaseError

An error occurred in the service while attempting a database operation.

ADServiceDatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

ADServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

FailedToConnectToDomainController

Contacting Active Directory failed.

FailedToGetOrganizationUnitInAD

Failed to access the OU in Active Directory.

FailedToGetDefaultComputerContainerInAD

Failed to access the default computers container in Active Directory.

FailedToCreateComputerAccountInAD

Failed to create the computer account in Active Directory.

FailedToAccessComputerAccountInAD

Failed to read the newly created computer account in Active Directory.

FailedToGetSidFromAD

Failed to get the SID for the created account from Active Directory.

FailedToSetSamAccountNameInAD



Failed to set the SAM account name in Active Directory for the account created.

FailedToSetUserAccountControlInAD

Failed to set the user account controller properties for the account created in Active Directory.

FailedToSaveChangeInAD

Failed to save the changes made to the created computer account in Active Directory.

FailedToSetPasswordInAD

Failed to set the password for the created computer account in Active Directory.

FailedToEnableAccountInAD

Failed to enable the newly created computer account in Active Directory.

ComputerNameAlreadyInUseInAD

The computer name for the computer to create is in use in Active Directory.

FailedToGetDistinguishedNameInAD

Failed to get the distinguished name for the created computer account in ActiveDirectory.

FailedToSetDnsHostNameInAD

Failed to set the Dns Host Name property for the created computer account in ActiveDirectory.

FailedToSetDisplayNameInAD

Failed to set the DisplayName property for the created computer account in ActiveDirectory.

FailedToWriteServicePrincipalNameInAD

Failed to set the ServicePrincipalName property for the created computer account in ActiveDirectory.

DiagnosticInformation <Exception>

Any other error information

SuccessfulAccounts <Citrix.AdIdentity.Sdk.Identity[]>

The list of accounts that were successfully added. Each object

provides details of the identity and contains the following information:

ADAccountSID <string>

The Sid of the identity.

ADAccountName <string>

The account name for the identity.

Domain <string>

The domain name that the account was created in.

State <string>

The current state of the AD account. This can be one of the following:

Error

The account is locked or disabled in AD.

Available

The account is in AD and available to be consumed by the other Machine Creation Services.

InUse

The account is in AD and is being consumed by the other Machine Creation Services.

Tainted

The account is in AD and no longer consumed by other Machine Creation Services. However, the password is no longer known so cannot be reused without 'Repairing' the account. See `repair-AcctADAccount` for details.

Lock <Boolean>

Indicates if the identity pool is locked.

Notes

In the case of failure, the following errors can result.

Error Codes

-----

NamingSchemeNotSpecifiedForIdentityPool

No naming scheme is defined in the specified identity pool.

IdentityPoolObjectNotFound

The specified identity pool was not located.

IdentityPoolAlreadyLocked

The specified identity pool is locked.

PermissionDenied

The user does not have administrative rights to perform this operation.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## CommunicationError

An error occurred while communicating with the service.

## ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>New-AcctADAccount -IdentityPoolName MyPool -Count 2 -OutVariable result
```

| SuccessfulAccounts                 | SuccessfulAccountsCount | FailedAccountsCount | FailedAccounts |
|------------------------------------|-------------------------|---------------------|----------------|
| {MyDomain\ACC001, MyDomain\ACC002} | 2                       | 0                   | {}             |

```
$result[0].SuccessfulAccounts
```

```
ADAccountSid : S-1-5-21-1315084875-1285793635-2418178940-2684
ADAccountName : MyDomain\ACC001
Domain       : MyDomain.com
State        : Available
Lock         : False
```

```
ADAccountSid : S-1-5-21-1315084875-1285793635-2418178940-2685
ADAccountName : MyDomain\ACC002
Domain       : MyDomain.com
State        : Available
Lock         : False
```

Creates two new AD accounts and registers them in the identity pool called "MyPool".

### ----- EXAMPLE 2 -----

```
c:\PS>New-AcctADAccount -IdentityPoolName MyPool -Count 2 -StartCount 50 -OutVariable result
```

| SuccessfulAccounts | SuccessfulAccountsCount | FailedAccountsCount | FailedAccounts |
|--------------------|-------------------------|---------------------|----------------|
|--------------------|-------------------------|---------------------|----------------|

```
-----  
{MyDomain\ACC050, MyDomain\ACC051} 2          0          {}
```

```
$result[0].SuccessfulAccounts
```

```
ADAccountSid : S-1-5-21-1315084875-1285793635-2418178940-2686
```

```
ADAccountName : MyDomain\ACC050
```

```
Domain      : MyDomain.com
```

```
State       : Available
```

```
Lock        : False
```

```
ADAccountSid : S-1-5-21-1315084875-1285793635-2418178940-2687
```

```
ADAccountName : MyDomain\ACC051
```

```
Domain      : MyDomain.com
```

```
State       : Available
```

```
Lock        : False
```

Creates two new AD accounts and registers them in the identity pool called "MyPool", starting from an index of 50.

# New-AcctIdentityPool

Jan 04, 2017

Creates a new identity pool.

## Syntax

```
New-AcctIdentityPool -IdentityPoolName <String> [-NamingScheme <String>] [-NamingSchemeType <ADIdentityNamingScheme>] [-OU <String>] [-Domain <String>] [-AllowUnicode] [-StartCount <Int32>] [-Scope <String[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to create identity pools that can be used to store AD computer accounts.

The naming scheme, naming scheme type, and domain must be specified if the identity pool is to be used to create new accounts.

Each identity pool is tied to a single domain. All the identities in an identity pool belong to the same domain. If the domain is not specified by a parameter to this command the domain will be set when an account is imported into it.

## Related topics

[Remove-AcctIdentityPool](#)

[Rename-AcctIdentityPool](#)

[Set-AcctIdentityPool](#)

[Test-AcctIdentityPoolNameAvailable](#)

[New-AcctADAccount](#)

## Parameters

**-IdentityPoolName**<String>

The name of the identity pool. This must not contain any of the following characters \/:#.\*?=<>|[]()''''

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-NamingScheme**<String>

Defines the template name for AD accounts created in the identity pool. The scheme can consist of fixed characters and a variable part defined by '#' characters. There can be only one variable region defined. The number of '#' characters defines the minimum length of the variable region. For example, a naming scheme of H#### could create accounts called H0001, H0002 (for a numeric scheme type) or HAAAA, HAAAB (for an alphabetic type).

Citrix recommends that you define a naming scheme that will not clash with accounts which already exist in AD; account creation will fail if a clash occurs.

There are restrictions on what constitutes a valid computer name: Minimum name length: 2 (DNS) Maximum name length: 15 bytes (NetBIOS) The following characters are not allowed in a computer name: backslash (\) slash mark (/) colon (:) asterisk (\*) question mark (?)

quotation mark (") less than sign (<) greater than sign (>) vertical bar (|) comma (,) tilde (~) exclamation point (!) at sign (@) number sign (#) dollar sign (\$) percent (%) caret (^) ampersand (&) apostrophe (') parenthesis (()) braces ({}), underscore (\_). The following names are reserved and must not be used at the end of the naming scheme: -GATEWAY -GW -TAC. Names must not start with a period (NetBIOS). Names must not be composed entirely of numbers (NetBIOS). Names must not contain a blank or space characters (DNS).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-NamingSchemeType**<ADIdentityNamingScheme>

The type of naming scheme. This can be Numeric or Alphabetic. This defines the format of the variable part of the AD account names that will be created.

|                        |         |
|------------------------|---------|
| Required?              | false   |
| Default Value          | Numeric |
| Accept Pipeline Input? | false   |

**-OU**<String>

The OU that computer accounts will be created into. If this is not specified, accounts are created into the default account container specified by AD. This is the 'Computers' container for out-of-the-box installations of AD. The OU must be a valid AD container and of the domain specified for the pool;

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Domain**<String>

The AD domain name for the pool. Specify this in FQDN format; for example, MyDomain.com.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AllowUnicode**<SwitchParameter>

Allow the naming scheme to have characters other than alphanumeric characters.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartCount**<Int32>

Defines the next number that will be used if creating new AD accounts for the identity pool.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Scope**<String[]>

The administration scopes to be applied to the new identity pool.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

Citrix.AdIdentity.Sdk.IdentityPool

This object provides details of the identity pool and contains the following information:

IdentityPoolName <string>

The name of the identity pool.

IdentityPoolUid <Guid>

The unique identifier for the identity pool.

NamingScheme <string>

The naming scheme for the identity pool.

NamingSchemeType <Citrix.XDInterServiceTypes.ADIIdentityNamingScheme>

The naming scheme type for the identity pool. This can be one of the following:

Numeric - naming scheme uses numeric indexes

Alphabetic - naming scheme uses alphabetic indexes

StartCount <int>

The next index to be used when creating an identity from the identity pool.

OU <string>

The Active Directory distinguished name for the OU in which accounts created from this identity pool will be created.

Domain <string>

The Active Directory domain that accounts in the pool belong to.

Lock <Boolean>

Indicates if the identity pool is locked.

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

UnableToConvertDomainName

Unable to convert domain name to DNS format.

NamingSchemeNotEnoughCharacters

Naming scheme does not have enough characters specified.

NamingSchemeTooManyCharacters



Naming scheme has too many characters specified.

NamingSchemeIllegalCharacter

Naming scheme contains illegal characters.

NamingSchemeMayNotStartWithPeriod

Naming scheme starts with a period (.) character.

NamingSchemeMayNotBeAllNumbers

Naming scheme contains only numbers.

NamingSchemeMissingNumericSpecifications

Naming scheme does not contain any variable specification (i.e. no '#' characters are specified).

NamingSchemeHasMoreThanOneSetOfHashes

Naming scheme has more than one variable region (i.e. there are '#' characters separated by other characters).

IdentityPoolDuplicateObjectExists

An identity pool with the same name already exists.

IdentityPoolOUInvalid

Identity Pool OU invalid as it does not exist.

IdentityPoolOUOfWrongDomain

IdentityPool OU invalid as it refers to a different domain to the domain specified for the pool.

InvalidIdentityPoolParameterCombination

Caused by either of the following validation errors:

\* If an OU is specified then a domain must also be specified.

\* NamingScheme, NamingSchemeType and Domain must all be present if any of these are specified.

NamingSchemeIllegalComputerName

The naming scheme supplied is not valid.

PermissionDenied

The user does not have administrative rights to perform this operation.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

CommunicationError

An error occurred while communicating with the service.

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>New-AcctIdentityPool -IdentityPoolName MyPool -NamingScheme Acc#### -Domain MyDomain.com -NamingSchemeType Numeric
```

```
IdentityPoolName : MyPool
IdentityPoolUid   : 22072d9e-6a8f-494b-a5bc-2ef18ca4b915
NamingScheme     : Acc####
NamingSchemeType : Numeric
StartCount       : 1
OU               :
Domain           : MyDomain.com
Lock             : False
```

Create a new identity pool from which accounts can be created in the domain called "MyDomain.com" using a numeric scheme of the format Acc####. The first account that will be created from this identity pool definition is Acc0001.

New AD accounts can be imported into this pool too, but must be from the Domain "MyDomain.com".

# Remove-AcctADAccount

Jan 04, 2017

Removes AD computer accounts from an identity pool.

## Syntax

```
Remove-AcctADAccount [-IdentityPoolName] <String> -ADAccountName <String[]> [-RemovalOption <ADIdentityRemoveAccountOption>] [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AcctADAccount [-IdentityPoolName] <String> -ADAccountSid <String[]> [-RemovalOption <ADIdentityRemoveAccountOption>] [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AcctADAccount -IdentityPoolUid <Guid> -ADAccountSid <String[]> [-RemovalOption <ADIdentityRemoveAccountOption>] [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AcctADAccount -IdentityPoolUid <Guid> -ADAccountName <String[]> [-RemovalOption <ADIdentityRemoveAccountOption>] [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove AD accounts from an identity pool. This removes the AD account from the Citrix service management scope. This process provides the options for removing the account in AD (or disabling it) if required.

All aspects of this command that need to make modifications to the accounts in AD will use the account that the runspace is using. This means that if an account is to be removed from AD or disabled, the user performing the operation in PowerShell must have sufficient privileges in AD for this operation to complete successfully.

If the option to remove the account from AD or to disable it in AD is specified, the AD operation must succeed for the account to be removed from the Citrix AD Identity Service database. Use caution when using the Force parameter because this allows removal of accounts that are in the 'inUse' state, which may result in the machines becoming unusable.

## Related topics

[New-AcctADAccount](#)

[Add-AcctADAccount](#)

[Repair-AcctADAccount](#)

[Unlock-AcctADAccount](#)

[Update-AcctADAccount](#)

[Get-AcctADAccount](#)

## Parameters

**-IdentityPoolName**<String>

The identity pool that the accounts are to be removed from.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ADAccountName<String[]>**

The AD account name to be removed. AD accounts are accepted in the following formats: Fully qualified DN e.g. CN=MyComputer,OU=Computers,DC=MyDomain,DC=Com; UPN format e.g MyComputer@MyDomain.Com; Domain qualified e.g MyDomain\MyComputer.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ADAccountSid<String[]>**

The Active Directory Account SID for the account to be removed.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IdentityPoolUid<Guid>**

The unique identifier for the identity pool that the accounts are to be removed from.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-RemovalOption<ADIdentityRemoveAccountOption>**

Defines the behavior relating to the AD account.

None - Do not attempt to remove the account from AD Delete - Attempt to remove the account from AD Disable - Attempt to disable the account in AD

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | None  |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Force**<SwitchParameter>

Indicates if accounts that are marked as 'in-use' can be removed.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Return Values

Citrix.AdIdentify.Sdk.AccountOperationSummary

The remove-AcctADAccount command returns an object with the following parameters:

SuccessfulAccountsCount <int>

The number of accounts that were removed successfully.

FailedAccountsCount <int>

The number of accounts that were not removed.

FailedAccounts <Citrix.AdIdentity.Sdk.AccountError[]>

The list of accounts that failed to be removed. Each one has the following parameters:

ADAccountName <string>

ADAccountSid <String>

ErrorReason <AdIdentityStatus>

This can be one of the following

UnableToConvertDomain

IdentityNotLocatedInDomain

IdentityNotInIdentityPool

IdentityObjectInUse

IdentityObjectLocked

ADServiceDatabaseError

ADServiceDatabaseNotConfigured

ADServiceStatusInvalidDb

FailedToConnectToDomainController

FailedToDisableAccountInAD

FailedToDeleteAccountInAD

FailedToExecuteSearchInAD

FailedToAccessComputerAccountInAD

DiagnosticInformation <Exception>

Any other error information

Notes

In the case of failure, the following errors can result.

Error Codes

-----

IdentityPoolNotFound

The specified identity pool was not found.

IdentityPoolAlreadyLocked

The specified identity pool was locked by another operation.

PermissionDenied

The user does not have administrative rights to perform this operation.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
C:\PS>Remove-AcctADAccount -IdentityPool MyPool -ADAccountName "Domain\account","domain\account2"
```

| SuccessfulAccountsCount | FailedAccountsCount | FailedAccounts |
|-------------------------|---------------------|----------------|
| -----                   | -----               | -----          |
| 2                       | 0                   | {}             |

Removes two accounts (account and account2) from the identity pool called "MyPool", leaving the AD accounts untouched.

##### ----- EXAMPLE 2 -----

```
C:\PS>Remove-AcctADAccount -IdentityPool MyPool -RemovalOption Delete -ADAccountName "Domain\account","domain\account2"
```

| SuccessfulAccountsCount | FailedAccountsCount | FailedAccounts |
|-------------------------|---------------------|----------------|
| -----                   | -----               | -----          |
| 2                       | 0                   | {}             |

Removes two accounts (account and account2) from the identity pool called "MyPool" (and from Active Directory).

##### ----- EXAMPLE 3 -----

```
C:\PS>Remove-AcctADAccount -IdentityPool MyPool -ADAccountName "Domain\account","domain\account2" -Force
```

| SuccessfulAccountsCount | FailedAccountsCount | FailedAccounts |
|-------------------------|---------------------|----------------|
| -----                   | -----               | -----          |
| 2                       | 0                   | {}             |

Removes two accounts (account and account2) from the identity pool called "MyPool", leaving the AD accounts untouched. The

accounts are removed regardless of whether they are in the 'inUse' state or not.

----- **EXAMPLE 4** -----

```
C:\PS>Remove-AcctADAccount -IdentityPool MyPool -ADAccountName "Domain\account","domain\account2" -OutVariable result
```

| SuccessfulAccountsCount | FailedAccountsCount | FailedAccounts |
|-------------------------|---------------------|----------------|
| 1                       | 1                   | {account2}     |

```
C:\PS>$result[0].FailedAccounts
```

| ADAccountName   | ADAccountSid | ErrorReason          |
|-----------------|--------------|----------------------|
| Domain\account2 | account2     | IdentityObjectLocked |

Shows failure of removal of one of two accounts and how to retrieve the failure reason.

----- **EXAMPLE 5** -----

```
C:\PS>Remove-AcctADAccount -IdentityPool MyPool -ADAccountSid S-1-5-21-1315084875-1285793635-2418178940-2685
```

| SuccessfulAccountsCount | FailedAccountsCount | FailedAccounts |
|-------------------------|---------------------|----------------|
| 1                       | 0                   | {}             |

Removes one account (S-1-5-21-1315084875-1285793635-2418178940-2685) from the identity pool called "MyPool", leaving the AD accounts untouched.



# Remove-AcctIdentityPool

Jan 04, 2017

Removes identity pools.

## Syntax

```
Remove-AcctIdentityPool [-IdentityPoolName] <String> [-LoggingId <Guid>] [-AdminAddress <String>]
```

```
[<CommonParameters>]
```

```
Remove-AcctIdentityPool -IdentityPoolUid <Guid> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove identity pools. The identity pool must be emptied of any AD accounts that it contains before it can be removed.

## Related topics

[New-AcctIdentityPool](#)

[Rename-AcctIdentityPool](#)

[Set-AcctIdentityPool](#)

[Unlock-AcctIdentityPool](#)

## Parameters

**-IdentityPoolName**<String>

The name of the identity pool to remove.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IdentityPoolUid**<Guid>

The unique identifier for the identity pool to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | LocalHost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Notes

In the case of failure the following errors can be produced.

#### Error Codes

-----

#### IdentityPoolObjectNotFound

The specified identity pool could not be located.

#### UnableToRemoveDueToAssociatedAccounts

The identity pool is not empty.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### ExceptionThrown

An unexpected error occurred. To locate more details see the Windows event logs on the controller being used, or examine the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
c:\Remove-AcctIdentityPool -IdentityPoolName MyPool  
Removes the identity pool called "MyPool".
```

# Remove-AcctIdentityPoolMetadata

Jan 04, 2017

Removes metadata from the given IdentityPool.

## Syntax

```
Remove-AcctIdentityPoolMetadata [-IdentityPoolUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AcctIdentityPoolMetadata [-IdentityPoolUid] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AcctIdentityPoolMetadata [-IdentityPoolName] <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AcctIdentityPoolMetadata [-IdentityPoolName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AcctIdentityPoolMetadata [-InputObject] <IdentityPool[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AcctIdentityPoolMetadata [-InputObject] <IdentityPool[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given IdentityPool.

## Related topics

[Set-AcctIdentityPoolMetadata](#)

## Parameters

**-IdentityPoolUid**<Guid>

Id of the IdentityPool

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-IdentityPoolName**<String>

Name of the IdentityPool

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<IdentityPool[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-

LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AcctIdentityPool | % { Remove-AcctIdentityPoolMetadata -Map $_.MetadataMap }
```

Remove all metadata from all IdentityPool objects.

# Remove-AcctIdentityPoolScope

Jan 04, 2017

Remove the specified IdentityPool(s) from the given scope(s).

## Syntax

```
Remove-AcctIdentityPoolScope [-Scope] <String[]> -InputObject <IdentityPool[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AcctIdentityPoolScope [-Scope] <String[]> -IdentityPoolUid <Guid[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AcctIdentityPoolScope [-Scope] <String[]> -IdentityPoolName <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The RemoveAcctIdentityPoolScope cmdlet is used to remove one or more IdentityPool objects from the given scope(s).

There are multiple parameter sets for this cmdlet, allowing you to identify the IdentityPool objects in different ways:

- IdentityPool objects can be piped in or specified by the InputObject parameter
- The IdentityPoolUid parameter specifies objects by IdentityPoolUid
- The IdentityPoolName parameter specifies objects by IdentityPoolName (supports wildcards)

To remove a IdentityPool from a scope you need permission to change the scopes of the IdentityPool.

If the IdentityPool is not in a specified scope, that scope will be silently ignored.

## Related topics

[Add-AcctIdentityPoolScope](#)

[Get-AcctScopedObject](#)

## Parameters

**-Scope**<String[]>

Specifies the scopes to remove the objects from.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-InputObject**<IdentityPool[]>

Specifies the IdentityPool objects to be removed.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-IdentityPoolUid**<Guid[]>

Specifies the IdentityPool objects to be removed by IdentityPoolUid.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-IdentityPoolName**<String[]>

Specifies the IdentityPool objects to be removed by IdentityPoolName.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

None

### Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### ScopeNotFound

One of the specified scopes was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command with the specified objects or scopes.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Remove-AcctIdentityPoolScope Finance -IdentityPoolUid 6702C5D0-C073-4080-A0EE-EC74CB537C52
```

Removes a single IdentityPool from the 'Finance' scope.

### ----- EXAMPLE 2 -----

```
c:\PS>Remove-AcctIdentityPoolScope Finance,Marketing -IdentityPoolUid 6702C5D0-C073-4080-A0EE-EC74CB537C52
```

Removes a single IdentityPool from multiple scopes.

### ----- EXAMPLE 3 -----

```
c:\PS>Get-AcctIdentityPool | Remove-AcctIdentityPoolScope Finance
```

Removes all visible IdentityPool objects from the 'Finance' scope.

### ----- EXAMPLE 4 -----

```
c:\PS>Remove-AcctIdentityPoolScope Finance -IdentityPoolName A*
```

Removes IdentityPool objects with a name starting with an 'A' from the 'Finance' scope.

# Remove-AcctServiceMetadata

Jan 04, 2017

Removes metadata from the given Service.

## Syntax

```
Remove-AcctServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AcctServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AcctServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AcctServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Service.

## Related topics

[Set-AcctServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Service[]>

Objects to which metadata is to be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Map<PSObject>**

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name<String>**

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId<Guid>**

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AcctService | % { Remove-AcctServiceMetadata -Map $_.MetadataMap }  
Remove all metadata from all Service objects.
```

# Rename-AcctIdentityPool

Jan 04, 2017

Renames an identity pool.

## Syntax

```
Rename-AcctIdentityPool [-IdentityPoolName] <String> [-NewIdentityPoolName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-AcctIdentityPool -IdentityPoolUid <Guid> -NewIdentityPoolName <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to change the name of an existing identity pool.

## Related topics

[Get-AcctIdentityPool](#)

[Set-AcctIdentityPool](#)

[Remove-AcctIdentityPool](#)

[Test-AcctIdentityPoolNameAvailable](#)

## Parameters

**-IdentityPoolName**<String>

The name of the identity pool to be renamed.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IdentityPoolUid**<Guid>

The unique identifier for the identity pool to be renamed.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-NewIdentityPoolName**<String>

The new name for the identity pool. This must be a name which is not used by an existing identity pool, and it must not contain any of the following characters \/:#.\*?=<>|[]()''''

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

Defines whether or not the command returns a result showing the new state of the updated provisioning scheme.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

Citrix.AdIdentity.Sdk.IdentityPool

This object provides details of the identity pool and contains the following information:

IdentityPoolName <string>

The name of the identity pool.

IdentityPoolUid <Guid>

The unique identifier for the identity pool.

NamingScheme <string>

The naming scheme for the identity pool.

NamingSchemeType <Citrix.XDInterServiceTypes.ADIdentityNamingScheme>

The naming scheme type for the identity pool. This can be one of the following:

Numeric - naming scheme uses numeric indexes

Alphabetic - naming scheme uses alphabetic indexes

StartCount <int>

The next index to be used when creating an identity from the identity pool.

OU <string>

The Active Directory distinguished name for the OU in which accounts created from this identity pool will be created.

Domain <string>

The Active Directory domain that accounts in the pool belong to.

Lock <Boolean>

Indicates if the identity pool is locked.

## Notes

In the case of failure the following errors can result.

### Error Codes

-----

IdentityPoolObjectNotFound

The specified identity pool could not be located.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
C:\PS>Rename-AcctIdentityPool -IdentityPoolName oldName -NewIdentityPoolName newName
```

Renames an existing identity pool called "oldName" to be called "newName".

# Repair-AcctADAccount

Jan 04, 2017

Resets the Active Directory machine password for the given accounts.

## Syntax

```
Repair-AcctADAccount -ADAccountName <String[]> [-Password <String>] [-SecurePassword <SecureString>] [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Repair-AcctADAccount -ADAccountSid <String[]> [-Password <String>] [-SecurePassword <SecureString>] [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This provides the ability to synchronize the account password stored in Active Directory with the password stored in the AD Identity Service. If successful, this results in the AD Identity Service account state being reset to 'available' so it can be consumed by other Machine Creation Services.

If the current account password is not supplied using the Password or SecurePassword Parameters, this requires the user who initiated the runspace to have the required permissions in Active Directory to reset the Active Directory Account password.

If the current account password is supplied then this command will use the password change operation which does not require any elevated permissions in Active Directory.

## Related topics

[New-AcctADAccount](#)

[Add-AcctADAccount](#)

[Remove-AcctADAccount](#)

[Unlock-AcctADAccount](#)

[Update-AcctADAccount](#)

[Get-AcctADAccount](#)

## Parameters

**-ADAccount Name**<String[]>

The Active Directory account name(s) that are to be repaired. Active Directory accounts are accepted in the following formats: Fully qualified DN e.g. CN=MyComputer,OU=Computers,DC=MyDomain,DC=Com; UPN format e.g. MyComputer@MyDomain.Com; Domain qualified e.g. MyDomain\MyComputer.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ADAccountSid**<String[]>

The Active Directory account SID(s) that are to be repaired.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Password**<String>

The current password for the computer account.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecurePassword**<SecureString>

The current password for the account (provided in a Secure String class).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Force**<SwitchParameter>

Indicates whether accounts that are marked as 'in-use' can be repaired or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

Citrix.AdIdentity.Sdk.AccountOperationSummary

The Repair-AcctADAccount command returns an object with the following parameters:

SuccessfulAccountsCount <int>

The number of accounts that were repaired successfully.

FailedAccountsCount <int>

The number of accounts that were not repaired.

FailedAccounts <Citrix.AdIdentity.Sdk.AccountError[]>

The list of accounts that failed to be repaired. Each one has the following parameters:

ADAccountName <string>

ADAccountSid <String>

ErrorReason <AdIdentityStatus>

This can be one of the following

UnableToConvertDomain

IdentityNotLocatedInDomain

IdentityNotFound

IdentityObjectInUse

IdentityObjectLocked

ADServiceDatabaseError

ADServiceDatabaseNotConfigured

ADServiceStatusInvalidDb

FailedToConnectToDomainController

FailedToExecuteSearchInAD

FailedToAccessComputerAccountInAD

FailedToSetPasswordInAD

FailedToChangePasswordInAD

DiagnosticInformation <Exception>

Any other error information

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with database failed for various reasons.

## CommunicationError

An error occurred while communicating with the service.

## ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS>Repair-AcctADAccount -ADAccountName "Domain\account","domain\account2"
```

```
SuccessfulAccountsCount  FailedAccountsCount  FailedAccounts
-----
                2                0 {}
```



# Reset-AcctServiceGroupMembership

Jan 04, 2017

Reloads the access permissions and configuration service locations for the AdIdentity Service.

## Syntax

```
Reset-AcctServiceGroupMembership [-ConfigServiceInstance] <ServiceInstance[]> [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

## Detailed Description

Enables you to reload AdIdentity Service access permissions and configuration service locations. The Reset-AcctServiceGroupMembership command must be run on at least one instance of the service type (Acct) after installation and registration with the configuration service. Without this operation, the AdIdentity services will be unable to communicate with other services in the XenDesktop deployment. When the command is run, the services are updated when additional services are added to the deployment, provided that the configuration service is not stopped. The Reset-AcctServiceGroupMembership command can be run again to refresh this information if automatic updates do not occur when new services are added to the deployment. If more than one configuration service instance is passed to the command, the first instance that meets the expected service type requirements is used.

## Related topics

### Parameters

**-ConfigServiceInstance**<ServiceInstance[]>

Specifies the configuration service instance object that represents the service instance for the type 'InterService' that references a configuration service for the deployment.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? |       |

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Input Type

Citrix.AdIdentity.Sdk.ServiceInstance[] Service instances containing a ServiceInstance object that refers to the central configuration service interservice interface can be piped to the Reset-AcctServiceGroupMembership command.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoSuitableServiceInstance

None of the supplied service instance objects were suitable for resetting service group membership.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-AcctServiceGroupMembership
```

Reset the service group membership for a service in a deployment where the configuration service is configured and running on the same machine as the service.

### ----- EXAMPLE 2 -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config -AdminAddress OtherServer.example.com | Reset-AcctServiceGroupmembership
```

Reset the service group membership for a service in a deployment where the configuration service that is configured and running on a machine named 'OtherServer.example.com'.

# Set-AcctDBConnection

Jan 04, 2017

Configures a database connection for the AdIdentity Service.

## Syntax

```
Set-AcctDBConnection [-DBConnection] <String> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Configures a connection to a database in which the AdIdentity Service can store its state. The service will attempt to connect and start using the database immediately after the connection is configured. The database connection string is updated to the specified value regardless of whether it is valid or not. Specifying an invalid connection string prevents a service from functioning until the error is corrected.

After a connection is configured, you cannot alter it without first clearing it (by setting the connection to \$null).

You do not need to configure a database connection to use this command.

## Related topics

[Get-AcctServiceStatus](#)

[Get-AcctDBConnection](#)

[Test-AcctDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be used by the AdIdentity Service. Passing in \$null will clear any existing database connection configured.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Force**<SwitchParameter>

If present, allows the local administrator to set the connection string to null when there are problems contacting the database or other services.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and

Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

#### Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Set-AcctDBConnection command returns an object containing the status of the AdIdentity Service together with extra diagnostics information.

DBUnconfigured

The AdIdentity Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the AdIdentity Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the AdIdentity Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the AdIdentity Service currently in use is incompatible with the version of the AdIdentity Service schema on the database. Upgrade the AdIdentity Service to a more recent version.

DBOlderVersionThanService

The version of the AdIdentity Service schema on the database is incompatible with the version of the AdIdentity Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The AdIdentity Service is running and is connected to a database containing a valid schema.

Failed

The AdIdentity Service has failed.

Unknown

The status of the AdIdentity Service cannot be determined.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidDBConnectionString

The database connection string has an invalid format.

DatabaseConnectionDetailsAlreadyConfigured

There was already a database connection configured. After a configuration is set, it can only be set to \$null.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-AcctDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Configures a database connection string for the AdIdentity Service.

----- **EXAMPLE 2** -----

```
c:\PS>Set-AcctDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Configures an invalid database connection string for the AdIdentity Service.

# Set-AcctIdentityPool

Jan 04, 2017

Update parameters of an identity pool.

## Syntax

```
Set-AcctIdentityPool [-IdentityPoolName] <String> [-NamingScheme <String>] [-NamingSchemeType  
<ADIdentityNamingScheme>] [-OU <String>] [-Domain <String>] [-AllowUnicode] [-PassThru] [-StartCount <Int32>] [-  
LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AcctIdentityPool -IdentityPoolUid <Guid> [-NamingScheme <String>] [-NamingSchemeType  
<ADIdentityNamingScheme>] [-OU <String>] [-Domain <String>] [-AllowUnicode] [-PassThru] [-StartCount <Int32>] [-  
LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to modify the parameters of an identity pool.

Note: When changing a naming scheme or naming scheme type, the index is not reset to 0; it continues to avoid AD account name clashes with existing accounts. If required, use the `New-AcctAdAccount` command to change the index when creating further accounts.

## Related topics

[New-AcctIdentityPool](#)

[Get-AcctIdentityPool](#)

[Remove-AcctIdentityPool](#)

## Parameters

**-IdentityPoolName**<String>

The name of the identity pool that is to be modified.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IdentityPoolUid**<Guid>

The unique identifier for the identity pool that is to be updated.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-NamingScheme**<String>

The new naming scheme that is to be used for the identity pool.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-NamingSchemeType**<ADIdentityNamingScheme>

The new naming scheme type that is to be used for the identity pool. This can be Numeric or Alphabetic.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OU**<String>

The new OU to be used for the Identity Pool. All accounts created after this is set are created in this AD container. This will not move any of the existing accounts. The OU must be a valid AD container and of the domain specified for the pool.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Domain**<String>

The new Active Directory domain that is to be used for the identity pool. All new accounts will be created in this domain, but this will not impact any of the existing accounts. The domain can be specified in either long or short form (i.e. domain or domain.com).

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AllowUnicode**<SwitchParameter>

Updates the definition of the allowed characters in a naming scheme.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

Defines whether the command returns the new state of the identity pool or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-StartCount**<Int32>

The start index for the next create operation

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | LocalHost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

Citrix.AdIdentity.Sdk.IdentityPool

This object provides details of the identity pool and contains the following information:

IdentityPoolName <string>

The name of the identity pool.

IdentityPoolUid <Guid>

The unique identifier for the identity pool.

NamingScheme <string>

The naming scheme for the identity pool.

NamingSchemeType <Citrix.XDInterServiceTypes.ADIdentityNamingScheme>

The naming scheme type for the identity pool. This can be one of the following:

Numeric - naming scheme uses numeric indexes

Alphabetic - naming scheme uses alphabetic indexes

StartCount <int>

The next index to be used when creating an identity from the identity pool.

OU <string>

The Active Directory distinguished name for the OU in which accounts created from this identity pool will be created.

Domain <string>

The Active Directory domain that accounts in the pool belong to.

Lock <Boolean>

Indicates whether the identity pool is locked.

## Notes

In the case of failure, the following errors can result.

## Error Codes

-----

### InvalidIdentityPoolParameterCombination

Caused by either of the following validation errors:

- \* If an OU is specified then a domain must also be specified.
- \* NamingScheme, NamingSchemeType and Domain must all be present if any of them are specified.

### NamingSchemeIllegalComputerName

The naming scheme supplied is not valid.

### UnableToConvertDomainName

Unable to convert domain name to DNS format.

### NamingSchemeNotEnoughCharacters

Naming scheme does not have enough characters specified.

### NamingSchemeTooManyCharacters

Naming scheme has too many characters specified.

### NamingSchemeIllegalCharacter

Naming scheme contains illegal characters.

### NamingSchemeMayNotStartWithPeriod

Naming scheme starts with a period (.) character.

### NamingSchemeMayNotBeAllNumbers

Naming scheme contains only numbers.

### NamingSchemeMissingNumericSpecifications

Naming scheme does not contain any variable specification (i.e. no '#' characters are specified).

#### NamingSchemeHasMoreThanOneSetOfHashes

Naming scheme has more than one variable region (i.e. there are '#' characters separated by other characters).

#### IdentityPoolDuplicateObjectExists

An identity pool with the same name exists already.

#### IdentityPoolObjectNotFound

The identity pool to be modified could not be located.

#### IdentityPoolOUInvalid

Identity Pool OU invalid as it does not exist.

#### IdentityPoolOUOfWrongDomain

IdentityPool OU invalid as it refers to a different domain to the domain specified for the pool.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
C:\PS>Set-AcctIdentityPool -IdentityPoolName poolName -StartCount 100 -NamingScheme AC####
```

Changes the start count, and the naming scheme, so that the next account generated will be AC0100 (assuming that account does not already exist)

# Set-AcctIdentityPoolMetadata

Jan 04, 2017

Adds or updates metadata on the given IdentityPool.

## Syntax

```
Set-AcctIdentityPoolMetadata [-IdentityPoolUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AcctIdentityPoolMetadata [-IdentityPoolUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AcctIdentityPoolMetadata [-IdentityPoolName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AcctIdentityPoolMetadata [-IdentityPoolName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AcctIdentityPoolMetadata [-InputObject] <IdentityPool[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AcctIdentityPoolMetadata [-InputObject] <IdentityPool[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given IdentityPool objects.

## Related topics

[Remove-AcctIdentityPoolMetadata](#)

## Parameters

**-IdentityPoolUid**<Guid>

Id of the IdentityPool

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-IdentityPoolName**<String>

Name of the IdentityPool

|                        |      |
|------------------------|------|
| Required?              | true |
| Default Value          |      |
| Accept Pipeline Input? |      |

|                        |                                |
|------------------------|--------------------------------|
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<IdentityPool[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the IdentityPool specified. The property cannot contain any of the following characters \/:;#.\*?=<> | [] () ""

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-AcctIdentityPoolMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.



#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
c:\PS>Set-AcctIdentityPoolMetadata -IdentityPoolUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Key      | Value |
|----------|-------|
| ---      | ----- |
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the IdentityPool with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Set-AcctServiceMetadata

Jan 04, 2017

Adds or updates metadata on the given Service.

## Syntax

```
Set-AcctServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AcctServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AcctServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AcctServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Allows you to store additional custom data against given Service objects.

## Related topics

[Remove-AcctServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Service[]>

Objects to which metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Service specified. The property cannot contain any of the following characters \/:#.\*?=<>|[]()''

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

System.Collections.Generic.Dictionary[String,String]

Set-AcctServiceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

**Notes**

If the command fails, the following errors can be returned.

**Error Codes**

-----

**InvalidParameterCombination**

The cmdlet parameters are inconsistent.

**UnknownObject**

One of the specified objects was not found.

**DatabaseError**

An error occurred in the service while attempting a database operation.

**DatabaseNotConfigured**

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Set-AcctServiceMetadata -ServiceHostId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Key      | Value |
|----------|-------|
| ---      | ----  |
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the Service with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Test-AcctDBConnection

Jan 04, 2017

Tests a database connection for the AdIdentity Service.

## Syntax

```
Test-AcctDBConnection [-DBConnection] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Tests a connection to the database in which the AdIdentity Service can store its state. The service will attempt to connect to the database without affecting the current connection to the database.

You do not have to clear the connection to use this command.

## Related topics

[Get-AcctServiceStatus](#)

[Get-AcctDBConnection](#)

[Set-AcctDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be tested by the AdIdentity Service.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Test-AcctDBConnection command returns an object containing the status of the AdIdentity Service if the connection string of the specified data store were to be set to the string being tested, together with extra diagnostics information for the specified connection string.

DBRejectedConnection

The database rejected the logon attempt from the AdIdentity Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the AdIdentity Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the AdIdentity Service currently in use is incompatible with the version of the AdIdentity Service schema on the database. Upgrade the AdIdentity Service to a more recent version.

DBOlderVersionThanService

The version of the AdIdentity Service schema on the database is incompatible with the version of the AdIdentity Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The Set-AcctDBConnection command would succeed if it were executed with the supplied connection string.

Failed

The AdIdentity Service has failed.

Unknown

The status of the AdIdentity Service cannot be determined.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidDBConnectionString

The database connection string has an invalid format.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Test-AcctDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Tests a database connection string for the AdIdentity Service.

----- **EXAMPLE 2** -----

```
c:\PS>Test-AcctDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Tests an invalid database connection string for the AdIdentity Service.



# Test-AcctIdentityPoolNameAvailable

Jan 04, 2017

Checks to ensure that the proposed name for an identity pool is unused.

## Syntax

```
Test-AcctIdentityPoolNameAvailable [-IdentityPoolName] <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Checks to ensure that the proposed name for an identity pool is unused. This check is done without regard for scoping of existing identity pools, so the names of inaccessible pools are also checked.

## Related topics

[New-AcctIdentityPool](#)

[Rename-AcctIdentityPool](#)

## Parameters

**-IdentityPoolName**<String[]>

The name or names of the identity pool(s) to be tested.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

object[]

An array of PSObjects that pair the name and availability of the name

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

Test-AcctIdentityPoolNameAvailable -IdentityPoolName \$NewPoolName

This tests whether the value of \$NewPoolName is unique or not, and can be used to create a new provisioning scheme or rename an existing one without failing. True is returned if the name is good.

# Unlock-AcctADAccount

Jan 04, 2017

Unlocks AD accounts within the AD Identity Service.

## Syntax

```
Unlock-AcctADAccount -ADAccountName <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Unlock-AcctADAccount -ADAccountSid <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to unlock the AD Identity Service identity item that references a specified AD account. An AD account is marked as locked in the AD Identity Service while the Machine Creation Services (MCS) are processing tasks relating to the account. If these tasks are forcibly stopped, an account can remain locked despite no longer being processed. This command resolves this issue, but use it with caution because unlocking an account that MCS expects to be locked can result in an MCS operation being cancelled. Use this command only when MCS has locked an account for use in a provisioning operation, and where this operation has failed without unlocking the account.

Note: This command does NOT make any changes to the account information stored in Active Directory.

## Related topics

[Get-AcctADAccount](#)

[New-AcctADAccount](#)

[Add-AcctADAccount](#)

[Repair-AcctADAccount](#)

[Remove-AcctADAccount](#)

[Update-AcctADAccount](#)

[Unlock-AcctADAccount](#)

## Parameters

**-ADAccountName**<String>

The AD account name to be unlocked. AD account name is accepted in the following formats: Fully qualified DN e.g. CN=MyComputer,OU=Computers,DC=MyDomain,DC=Com; UPN format e.g MyComputer@MyDomain.Com; Domain qualified e.g MyDomain\MyComputer.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-ADAccountSid**<String>

The AD account SID that represents the account to be unlocked.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | LocalHost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Input Type

Citrix.AdIdentity.Sdk.IdentityInPool You can pipe an object containing a parameter called 'ADAccountSID' to unlock-AcctADAccount.

### Notes

In the case of failure, the following errors can result.

### Error Codes

-----

IdentityNotLocatedInDomain

The specified AD account could not be located in Active Directory.

IdentityObjectNotFound

The identity could not be found.

IdentityAlreadyUnlocked

The identity is not locked.

PermissionDenied

The user does not have administrative rights to perform this operation.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for for various reasons.

CommunicationError

An error occurred while communicating with the service.

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Unlock-AcctADAccount -ADAccountName Domain\account
Unlocks the AD account called "Domain\account".
```

----- **EXAMPLE 2** -----

```
c:\PS>Get-AcctADAccount -Filter {lock -eq $true} | Unlock-AcctADAccount
Unlocks all the locked AD accounts.
```



# Unlock-AcctIdentityPool

Jan 04, 2017

Unlocks identity pools.

## Syntax

```
Unlock-AcctIdentityPool [-IdentityPoolName] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Unlock-AcctIdentityPool -IdentityPoolUid <Guid> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to unlock the specified identity pool. Identity pools are locked automatically when being updated (e.g. when new accounts are being created into them). The pool must never be left in a locked state; this command allows recovery from an error should this ever occur. Use this command with caution, as unlocking an identity pool which is supposed to be locked may result in unexpected behavior.

## Related topics

[New-AcctIdentityPool](#)

[Remove-AcctIdentityPool](#)

[Set-AcctIdentityPool](#)

## Parameters

**-IdentityPoolName**<String>

The name of the identity pool to unlock.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IdentityPoolUid**<Guid>

The unique identifier for the identity pool to be unlocked.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Input Type

Citrix.AdIdentity.Sdk.IdentityPool You can pipe an object containing a parameter called 'IdentityPoolName' to unlock-AcctIdentityPool.

### Notes

In the case of failure, the following errors can result.

#### Error Codes

-----

#### IdentityPollObjectNotFound

The specified identity pool could not be located.

#### IdentityPoolAlreadyUnlocked

The specified identity pool is not locked.

#### PermissionDenied

The user does not have administrative rights to perform this operation.



## ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## CommunicationError

An error occurred while communicating with the service.

## ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS>Unlock-AcctIdentityPool -IdentityPool MyPool  
Unlocks the identity pool called "MyPool".
```

### ----- EXAMPLE 2 -----

```
C:\PS>Get-AcctIdentityPool -Filter {Lock -eq $true} | Unlock-AcctIdentityPool  
Unlocks all the locked identity pools.
```

# Update-AcctADAccount

Jan 04, 2017

Refreshes the AD computer account state stored in the AD Identity Service.

## Syntax

```
Update-AcctADAccount [-IdentityPoolName] <String> [-AllAccounts] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Update-AcctADAccount -IdentityPoolUid <Guid> [-AllAccounts] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to synchronize the state of the AD accounts stored in the AD Identity Service with the AD accounts themselves. By default, this checks all accounts marked as 'error' to determine if accounts are still in an error state (i.e. disabled or locked). If you specify the 'AllAccounts' option, it checks accounts not in error state and updates the status of these accounts too.

## Related topics

[New-AcctADAccount](#)

[Add-AcctADAccount](#)

[Remove-AcctADAccount](#)

[Repair-AcctADAccount](#)

[Unlock-AcctADAccount](#)

[Get-AcctADAccount](#)

## Parameters

**-IdentityPoolName**<String>

The name of the identity pool for the accounts to be refreshed.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IdentityPoolUid**<Guid>

The unique identifier for the identity pool of the AD accounts that are to be refreshed.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AllAccounts**<SwitchParameter>

Indicates if all accounts should be refreshed or only the ones marked as in error.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | LocalHost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Notes

In the case of failure, the following errors can result.

## Error Codes

-----

### PermissionDenied

The user does not have administrative rights to perform this operation.

### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with database failed for various reasons.

### CommunicationError

An error occurred while communicating with the service.

### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Update-AcctADAccount -IdentityPoolName MyPool
```

Checks the status of all accounts in the Identity Pool MyPool that are currently mark as in the Error state and checks if the account is now available.

### ----- EXAMPLE 2 -----

```
c:\PS>Update-AcctADAccount -IdentityPoolName MyPool -AllAccounts
```

Checks the status of all accounts in the Identity Pool MyPool marking them as Available, InUse, Tainted or Error as appropriate.

# Citrix.AppV.Admin.V1

Jan 04, 2017

## Cmdlets

| Name                                         | Description                                                                                                                                                                                                                                |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">ConvertTo-CtxAppVLauncherArg</a> | Returns a string containing information to send to the App-V Launcher. You can plug this string directly into the Virtual Delivery Agent (VDA) to launch App-V applications.                                                               |
| <a href="#">Get-CtxAppVApplication</a>       | Enumerates all published App-V applications for a given Management server.                                                                                                                                                                 |
| <a href="#">Get-CtxAppVApplicationInfo</a>   | Enumerates information for a given application in a given package for a given Management server.                                                                                                                                           |
| <a href="#">Get-CtxAppVServer</a>            | Returns URLs for App-V Publishing and Management servers contained in a Citrix App-V policy. Returned values are in string format.                                                                                                         |
| <a href="#">Get-CtxAppVServerSetting</a>     | Returns settings for the specified App-V Publishing Server.                                                                                                                                                                                |
| <a href="#">New-CtxAppVServer</a>            | Creates a new Citrix App-V policy containing the specified App-V Management and Publishing Server URLs.                                                                                                                                    |
| <a href="#">Set-CtxAppVServerSetting</a>     | Specifies the App-V Publishing server settings to use on the VDA. These settings determine whether or not the App-V Client can automatically initiate a publishing refresh on certain events such as user logon or at specified intervals. |
| <a href="#">Test-CtxAppVServer</a>           | Tests the given URL for the presence of App-V Management and Publishing servers.                                                                                                                                                           |

# ConvertTo-CtxAppVLauncherArg

Jan 04, 2017

Returns a string containing information to send to the App-V Launcher. You can plug this string directly into the Virtual Delivery Agent (VDA) to launch App-V applications.

## Syntax

ConvertTo-CtxAppVLauncherArg [-AppVPublishingServer] <string> [[-PackageId] <String>] [-AppId] <String> -SeqLoc <String> -TargetInPackage <boolean> [<CommonParameters>]

ConvertTo-CtxAppVLauncherArg [-LauncherPath] <SwitchParameter> [<CommonParameters>]

## Detailed Description

Returns a string containing information to send to the App-V Launcher. You can plug this string directly into the Virtual Delivery Agent (VDA) to launch App-V applications.

## Related topics

## Parameters

### -AppVPublishingServer<>

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

### -PackageId<>

The Package Id of the application.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | false                          |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

### -AppId<>

The App Id of the application.

|                        |       |
|------------------------|-------|
| Required?              | True  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### -SeqLoc<>

The sequence location of the package.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### -TargetInPackage<>

Pass this as \$true if TargetInPackage field is "true" in the Manifest file for the particular App Id.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### -LauncherPath<>

Gets the Citrix Launcher path. The Citrix Launcher is a component installed on the VDA.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

Examples

----- **EXAMPLE 1** -----

ConvertTo-CtxAppVLauncherArg -AppVPublishingServer "http://appv-2k82-srv.blrstrm.com:8082" -PackageId "1bcb6993-10b1-4659-b9d4-e809e10cecdf\_5" -Appld "[[ProgramFilesX86]]Beyond Compa  
 Converts the arguments provided into a cmdlet to launch Beyond Compare from http://appv-server on the VDA.

----- **EXAMPLE 2** -----

ConvertTo-CtxAppVLauncherArg -LauncherPath  
 Gets the Citrix App-V Launcher path.

# Get-CtxAppVApplication

Jan 04, 2017

Enumerates all published App-V applications for a given Management server.

## Syntax

```
Get-CtxAppVApplication [-AppVManagementServer] <string> [<CommonParameters>]
```

## Detailed Description

Queries a given Management server and fetches all published applications for that server. Applications that are published but have no user Entitlements are not displayed by this cmdlet.

## Related topics

## Parameters

**-AppVManagementServer**<String>

App-V Management Server

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

## Examples

----- **EXAMPLE 1** -----

```
Get-CtxAppVApplication -AppVManagementServer "xmas-demo-appv"
```

Displays all published applications for the Management server "xmas-demo-appv" .



# Get-CtxAppVApplicationInfo

Jan 04, 2017

Enumerates information for a given application in a given package for a given Management server.

## Syntax

```
Get-CtxAppVApplicationInfo [-AppVManagementServer] <string> [-AppId] <String> [-PackageId] <string> [[-Property] <string[]>] [<CommonParameters>]
```

```
Get-CtxAppVApplicationInfo [-AppVManagementServer] <String> [[-InputObject] <AppVServeApplications[]>] [[-Property] <string[]>] [<CommonParameters>]
```

## Detailed Description

Queries a given Management server and fetches the requested information for a given application.

## Related topics

## Parameters

**-AppVManagementServer**<string>

Machine name of the App-V Management server. The name does not need to be specified as a Fully Qualified Domain Name (FQDN).

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AppId**<string>

The App Id of the given application.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PackageId**<string>

The Package Id of the given application.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## Return Values

Citrix.VirtApp.Studio.PowerShellManager.AppVAppData

## Examples

----- **EXAMPLE 1** -----

```
Get-CtxAppVApplicationInfo -AppVManagementServer "xmas-demo-appv" -AppId "[(ProgramFilesX86)]Beyond Compare 3\BCompare.exe" -PackageId "1bcb6993-10b1-4659-b9d4-e809e10cecdf_5" -f  
Fetches all application information for the Beyond Compare 3 application from the xmas-demo-appv Management server.
```

----- **EXAMPLE 2** -----

```
Get-CtxAppVApplicationInfo -AppVManagementServer "xmas-demo-appv" -AppId "[(ProgramFilesX86)]Beyond Compare 3\BCompare.exe" -PackageId "1bcb6993-10b1-4659-b9d4-e809e10cecdf_5" -f  
Fetches only information about FTA and User Entitlements for the Beyond Compare 3 application from the xmas-demo-appv Management server.
```

----- **EXAMPLE 3** -----

```
Get-CtxAppVApplication -AppVManagementServer "xmas-demo-appv" | Get-CtxAppVApplicationInfo -AppVManagementServer "xmas-demo-appv"  
Pipelines the output of Get-CtxAppVApplication to Get-CtxAppVApplicationInfo to get the application properties of all the published applications on the xmas-demo-appv Management server.
```

# Get-CtxAppVServer

Jan 04, 2017

Returns URLs for App-V Publishing and Management servers contained in a Citrix App-V policy. Returned values are in string format.

## Syntax

```
Get-CtxAppVServer -ByteArray <byte[]> [-ConsumedByStudio <bool>] [<CommonParameters>]
```

## Detailed Description

Returns URLs for App-V Publishing and Management servers contained in a Citrix App-V policy. Returned values are in string format.

## Related topics

## Parameters

### **-ByteArray**<>

Specifies the Citrix App-V policy created using the New-CtxAppVServer cmdlet.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

### **-ConsumedByStudio**<>

If set to "true", outputs URLs for the App-V Publishing and Management servers. If set to "false", outputs both the URL and settings for the App-V Publishing server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## Examples

----- **EXAMPLE 1** -----

```
$config = Get-BrokerMachineConfiguration -Name appv*
```

```
Get-CtxAppVServer -ByteArray $config[0].Policy
```

Returns Publishing Server URL , Management Server URL configured in given policy



# Get-CtxAppVServerSetting

Jan 04, 2017

Returns settings for the specified App-V Publishing Server.

## Syntax

```
Get-CtxAppVServerSetting -AppVPublishingServer <string> [<CommonParameters>]
```

## Detailed Description

Returns settings for the specified App-V Publishing Server. For more information about these settings, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

## Related topics

## Parameters

### **-AppVPublishingServer**<>

URL of the Publishing Server for which settings are to be returned.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

## Return Values

Publishing Server settings. These settings are : GlobalRefreshEnabled;  
GlobalRefreshOnLogon;GlobalRefreshInterval;GlobalRefreshIntervalUnit;UserRefreshEnabled;  
UserRefreshOnLogon;UserRefreshInterval;UserRefreshIntervalUnit;

## Examples

### ----- EXAMPLE 1 -----

```
Get-CtxAppVServerSetting -AppVPublishingServer http://AppV50Publishing:8082
```

This example returns settings associated with <http://AppV50PublishingServer:8082> in the following format:

GlobalRefreshEnabled: false

GlobalRefreshOnLogon: false

GlobalRefreshInterval: 0

GlobalRefreshIntervalUnit: Day

UserRefreshEnabled: true

UserRefreshOnLogon: false

UserRefreshInterval: 0

UserRefreshIntervalUnit: Hour

# New-CtxAppVServer

Jan 04, 2017

Creates a new Citrix App-V policy containing the specified App-V Management and Publishing Server URLs.

## Syntax

```
New-CtxAppVServer -PublishingServer <string> -ManagementServer <String> [-UserRefreshEnabled [<Boolean>]] [-UserRefreshOnLogon <Boolean>] [-UserRefreshInterval <int>] [-UserRefreshIntervalUnit <Enum>] [-GlobalRefreshEnabled [<Boolean>]] [-GlobalRefreshOnLogon <Boolean>] [-GlobalRefreshInterval <Int>] [-GlobalRefreshIntervalUnit <Enum>] [-CommonParameters< >]
```

## Detailed Description

Creates a new Citrix App-V policy containing the specified App-V Management and Publishing server URLs. Additionally, accepts Publishing Server settings that control how and when automatic refresh occurs on the VDA.

## Related topics

### Parameters

#### -PublishingServer<>

URL of the Publishing server to add to the Citrix policy.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

#### -ManagementServer<>

URL of the Management server to add to the Citrix policy.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

#### -UserRefreshEnabled<>

Enables a refresh of packages published to user groups either at user logon or at a specified interval. For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### -UserRefreshOnLogon<>

Specifies whether or not to initiate a refresh of packages published to user groups on every user logon. For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### -UserRefreshInterval<>

Specifies the frequency at which to initiate a refresh of packages published to user groups. This can be either days or hours, as specified by the UserRefreshIntervalUnit setting. For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-UserRefreshInterval<>**

Specifies the frequency at which to initiate a refresh of packages published to user groups. This can be either days or hours, as specified by the UserRefreshIntervalUnit setting. For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserRefreshIntervalUnit<>**

Specifies the unit for the UserRefreshInterval setting. This can be set to either Hours (0) or Days (1). For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-GlobalRefreshEnabled<>**

Enables a refresh of packages published to machine groups either at user logon or at a specified interval. For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-GlobalRefreshOnLogon<>**

Specifies whether or not to initiate a refresh of packages published to machine groups on every user logon. For more information, see the App-V 5.0 documentation at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-GlobalRefreshInterval<>**

Specifies the frequency at which to initiate a refresh of packages published to machine groups. This can be either days or hours, as specified by the GlobalRefreshIntervalUnit setting. Please refer to App-V 5.0 documentation for details. <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-GlobalRefreshIntervalUnit<>**

Specifies the unit for the GlobalRefreshInterval setting. This can be set to either Hours (0) or Days (1). Please refer to App-V 5.0 documentation for details. <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## Examples

### EXAMPLE 1

New-CtxAppVServer -ManagementServer http://AppV-Mgmt-Server:8080 -PublishingServer http://appV-Pub-Server:8082

Creates a new Citrix Policy for Management Server AppV-Mgmt-Server:8080 & Publishing Server: AppV-Mgmt-Server:8082. Default Publishing Server setting is used for http://AppV-Mgmt-Server:8082

Default values for publishing server settings used for Http://AppV-Mgmt-Server:8082 are:

GlobalRefreshEnabled = false; GlobalLogonRefresh = false ; GlobalIntervalRefreshInterval = 0; GlobalIntervalRefreshUnit = Day

UserRefreshEnabled = true; UserLogonRefresh = true ; UserIntervalRefreshInterval = 0; GlobalIntervalRefreshUnit = Day

### EXAMPLE 1

New-CtxAppVServer -ManagementServer http://AppV-Mgmt-Server:8080 -PublishingServer http://appV-Pub-Server:8082 -GlobalRefreshEnabled \$true -GlobalLogonRefresh \$true -GlobalRefreshInterval  
Creates a new Citrix Policy for Management Server AppV-Mgmt-Server:8080 & Publishing Server: AppV-Mgmt-Server:8082. User specified Publishing Server settings are used for AppV-Mgmt-Server:8082

Following values are used to configure Publishing Server : http://AppV-Mgmt-Server:8082

GlobalRefreshEnabled = True; GlobalLogonRefresh = True ; GlobalIntervalRefreshInterval = 2; GlobalIntervalRefreshUnit = Hour

UserRefreshEnabled = true; UserLogonRefresh = true ; UserIntervalRefreshInterval = 3; GlobalIntervalRefreshUnit = Hour



# Set-CtxAppVServerSetting

Jan 04, 2017

Specifies the App-V Publishing server settings to use on the VDA. These settings determine whether or not the App-V Client can automatically initiate a publishing refresh on certain events such as user logon or at specified intervals.

## Syntax

```
Set-CtxAppVServerSetting -AppVPublishingServer <string> [-UserRefreshEnabled [<Boolean>]] [-UserRefreshOnLogon <Boolean>] [-UserRefreshInterval <int>] [-UserRefreshIntervalUnit <Enum>] [-GlobalRefreshEnabled [<Boolean>]] [-GlobalRefreshOnLogon <Boolean>] [-GlobalRefreshInterval <int>] [-GlobalRefreshIntervalUnit <Enum>] [<CommonParameters>]
```

## Detailed Description

Specifies the App-V Publishing server settings to use on the VDA. These settings determine whether or not the App-V Client can automatically initiate a publishing refresh on certain events such as user logon or at specified intervals. Please refer to Microsoft App-V 5.0 documentation for more details on these settings : <http://technet.microsoft.com/en-us/library/jj687745.aspx>

## Related topics

### Parameters

#### -AppVPublishingServer<>

URL of the Publishing Server for which settings are to be set.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

#### -UserRefreshEnabled<>

Enables a refresh of packages published to user groups either at user logon or at a specified interval. For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | True  |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### -UserRefreshOnLogon<>

Specifies whether or not to initiate a refresh of packages published to user groups on every user logon. For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### -UserRefreshInterval<>

Specifies the frequency at which to initiate a refresh of packages published to user groups. This can be either days or hours, as specified by the UserRefreshIntervalUnit setting. For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### -UserRefreshIntervalUnit<>

Specifies the unit for the UserRefreshInterval setting. This can be set to either Hours (0) or Days (1). For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-GlobalRefreshEnabled<>**

Enables a refresh of packages published to machine groups either at user logon or at a specified interval. For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | True  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-GlobalRefreshOnLogon<>**

Specifies whether or not to initiate a refresh of packages published to machine groups on every user logon. For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-GlobalRefreshInterval<>**

Specifies the frequency at which to initiate a refresh of packages published to machine groups. This can be either days or hours, as specified by the GlobalRefreshIntervalUnit setting. For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-GlobalRefreshIntervalUnit<>**

Specifies the unit for the GlobalRefreshInterval setting. This can be set to either Hours (0) or Days (1). For more information, see the App-V documentation on the Microsoft website at <http://technet.microsoft.com/en-us/library/jj687745.aspx>

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

Examples

----- **EXAMPLE 1** -----

Set-CtxAppVServerSetting -AppVPublishingServer http://AppV50Publishing:8082 -GlobalRefreshEnabled \$true -GlobalRefreshOnLogon \$true -GlobalRefreshInterval 2 -GlobalRefreshIntervalUnit Hour -

# Test-CtxAppVServer

Jan 04, 2017

Tests the given URL for the presence of App-V Management and Publishing servers.

## Syntax

```
Test-CtxAppVServer [-AppVManagementServer] <string> [<CommonParameters>]
```

```
Test-CtxAppVServer [-AppVPublishingServer] <string> [<CommonParameters>]
```

## Detailed Description

Tests the given URL for the presence of App-V Management and Publishing servers.

## Related topics

## Parameters

**-AppVManagementServer**<String>

Machine name of the App-V Management server.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AppVPublishingServer**<String>

Machine name of the App-V Publishing server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## Return Values

Microsoft.AppvAgent.AppvClientPackage

## Examples

----- **EXAMPLE 1** -----

```
Test-CtxAppVServer -AppVManagementServer "xmas-demo-appv"
```

Tests whether "xmas-demo-appv" is a Management server or not. The name does not need to be specified as a Fully Qualified Domain Name (FQDN).

----- **EXAMPLE 2** -----

Test-CtxAppVServer -AppVPublishingServer "http://appvb2refserver.blrstrm.com:8082"

Tests whether "http://appvb2refserver.blrstrm.com:8082" is a Publishing server or not. Specify the full address, including FQDN and port number, of the Publishing server.

# Citrix.Broker.Admin.V2

Jan 04, 2017  
Overview

| Name                                               | Description                                                                |
|----------------------------------------------------|----------------------------------------------------------------------------|
| <a href="#">Broker AccessPolicy</a>                | Controls client-connection-based access to desktop groups.                 |
| <a href="#">Broker Applications</a>                | Describes how to publish and manage hosted applications.                   |
| <a href="#">Broker AssignmentPolicy</a>            | Controls the automatic, permanent assignment of machines to users.         |
| <a href="#">Broker Concepts</a>                    | Overview of the Citrix Broker.                                             |
| <a href="#">Broker ConfigurationSlots</a>          | Overview of assigning a collection of related settings to a desktop group. |
| <a href="#">Broker ControllerDiscovery</a>         | Describes the way that machines providing published resources discover     |
| <a href="#">Broker Desktops</a>                    | Describes desktop concepts and usage.                                      |
| <a href="#">Broker EntitlementPolicy</a>           | Controls end-user entitlement to desktop and application sessions provided |
| <a href="#">Broker ErrorHandling</a>               | Describes broker errors generated by cmdlets and how to access them.       |
| <a href="#">Broker Filtering</a>                   | Describes the common filtering options for XenDesktop cmdlets.             |
| <a href="#">Broker Licensing</a>                   | Overview of broker licensing configuration.                                |
| <a href="#">Broker Machines</a>                    | Describes machine concepts and usage.                                      |
| <a href="#">Broker Policies</a>                    | Overview of the site policies that control users' access to desktop and    |
| <a href="#">Broker PostInstallPreConfiguration</a> | Describes how to configure the Citrix Broker Service port numbers, URL     |
| <a href="#">Broker PowerManagement</a>             | Describes power management of machines used for desktops and applications. |
| <a href="#">Broker RemotePC</a>                    | Overview of the Remote PC feature.                                         |

## Cmdlets

| Name                                               | Description                                                              |
|----------------------------------------------------|--------------------------------------------------------------------------|
| <a href="#">Add-BrokerApplication</a>              | Adds applications to a desktop group.                                    |
| <a href="#">Add-BrokerDesktopGroup</a>             | Associate Remote PC desktop groups with the specified Remote PC catalog. |
| <a href="#">Add-BrokerMachine</a>                  | Adds one or more machines to a desktop group.                            |
| <a href="#">Add-BrokerMachineConfiguration</a>     | Adds a machine configuration to a desktop group.                         |
| <a href="#">Add-BrokerMachinesToDesktopGroup</a>   | Adds machines from a catalog to a desktop group.                         |
| <a href="#">Add-BrokerScope</a>                    | Add the specified catalog/desktop group to the given scope(s).           |
| <a href="#">Add-BrokerTag</a>                      | Associate a tag with another object.                                     |
| <a href="#">Add-BrokerUser</a>                     | Creates an association between a user and another broker object          |
| <a href="#">Disconnect-BrokerSession</a>           | Disconnect a session.                                                    |
| <a href="#">Export-BrokerDesktopPolicy</a>         | Gets the site wide Citrix Group Policy settings.                         |
| <a href="#">Get-BrokerAccessPolicyRule</a>         | Gets rules from the site's access policy.                                |
| <a href="#">Get-BrokerAdminFolder</a>              | Get the admin folders in this site.                                      |
| <a href="#">Get-BrokerAppAssignmentPolicyRule</a>  | Gets application rules from the site's assignment policy.                |
| <a href="#">Get-BrokerAppEntitlementPolicyRule</a> | Gets application rules from the site's entitlement policy.               |
| <a href="#">Get-BrokerApplication</a>              | Get the applications published on this site.                             |
| <a href="#">Get-BrokerApplicationInstance</a>      | Gets the running applications on the desktops.                           |
| <a href="#">Get-BrokerAssignmentPolicyRule</a>     | Gets desktop rules from the site's assignment policy.                    |
| <a href="#">Get-BrokerCatalog</a>                  | Gets catalogs configured for this site.                                  |

| Name                                | Description                                                                                  |
|-------------------------------------|----------------------------------------------------------------------------------------------|
| Get-BrokerConfigurationSlot         | Gets configuration slots configured for this site.                                           |
| Get-BrokerConfiguredFTA             | Gets any file type associations configured for an application.                               |
| Get-BrokerConnectionLog             | Get entries from the site's session connection log.                                          |
| Get-BrokerController                | Gets Controllers running broker services in the site.                                        |
| Get-BrokerDBConnection              | Gets the database connection string for the specified data store used by the Broker Service. |
| Get-BrokerDBSchema                  | Gets SQL scripts to create or maintain the database schema for the Citrix Broker Service.    |
| Get-BrokerDBVersionChangeScript     | Gets an SQL service schema update script for the Citrix Broker Service.                      |
| Get-BrokerDelayedHostingPowerAction | Gets power actions that are executed after a delay.                                          |
| Get-BrokerDesktop                   | Gets desktops configured for this site.                                                      |
| Get-BrokerDesktopGroup              | Gets broker desktop groups configured for this site.                                         |
| Get-BrokerDesktopUsage              | Get usage history of desktop groups.                                                         |
| Get-BrokerEntitlementPolicyRule     | Gets desktop rules from the site's entitlement policy.                                       |
| Get-BrokerHostingPowerAction        | Gets power actions queued for machines.                                                      |
| Get-BrokerHypervisorAlert           | Gets hypervisor alerts recorded by the controller.                                           |
| Get-BrokerHypervisorConnection      | Gets hypervisor connections matching the specified criteria.                                 |
| Get-BrokerIcon                      | Get stored icons.                                                                            |
| Get-BrokerImportedFTA               | Gets the imported file type associations.                                                    |
| Get-BrokerInstalledDbVersion        | Gets a list of all available database schema versions for the Broker Service.                |
| Get-BrokerLease                     | Gets stored leases.                                                                          |

| Name                                                   | Description                                                           |
|--------------------------------------------------------|-----------------------------------------------------------------------|
| <a href="#">Get-BrokerMachine</a>                      | Gets machines belonging to this site.                                 |
| <a href="#">Get-BrokerMachineCommand</a>               | Get the list of commands queued for delivery to a desktop.            |
| <a href="#">Get-BrokerMachineConfiguration</a>         | Gets machine configurations defined for this site.                    |
| <a href="#">Get-BrokerMachineStartMenuShortcutIcon</a> | Retrieves a Start Menu Shortcut icon from the specified machine.      |
| <a href="#">Get-BrokerMachineStartMenuShortcuts</a>    | Retrieves the Start Menu Shortcuts from the specified machine.        |
| <a href="#">Get-BrokerPowerTimeScheme</a>              | Gets power management time schemes for desktop groups.                |
| <a href="#">Get-BrokerPrivateDesktop</a>               | Get private desktops configured for this site.                        |
| <a href="#">Get-BrokerRebootCycle</a>                  | Gets one or more reboot cycles.                                       |
| <a href="#">Get-BrokerRebootSchedule</a>               | Gets one or more reboot schedules.                                    |
| <a href="#">Get-BrokerRemotePCAccount</a>              | Get RemotePCAccount entries configured for this site.                 |
| <a href="#">Get-BrokerResource</a>                     | Gets resources that a user can broker connections to.                 |
| <a href="#">Get-BrokerScopedObject</a>                 | Gets the details of the scoped objects for the Broker Service.        |
| <a href="#">Get-BrokerServiceAddedCapability</a>       | Gets any added capabilities for the Broker Service on the controller. |
| <a href="#">Get-BrokerServiceInstance</a>              | Gets the service instance entries for the Broker Service.             |
| <a href="#">Get-BrokerServiceStatus</a>                | Gets the current state of the Broker Service on the controller.       |
| <a href="#">Get-BrokerSession</a>                      | Gets a list of sessions.                                              |
| <a href="#">Get-BrokerSessionLinger</a>                | Gets one or more session lingering settings.                          |
| <a href="#">Get-BrokerSessionPreLaunch</a>             | Gets one or more session pre-launch settings.                         |
| <a href="#">Get-BrokerSharedDesktop</a>                | Get shared desktops configured for this site.                         |



| Get-BrokerSite<br>Name             | Description                                                                 |
|------------------------------------|-----------------------------------------------------------------------------|
| Get-BrokerTag                      | Gets one or more tags.                                                      |
| Get-BrokerUnconfiguredMachine      | Gets machines that have registered but are not yet configured in this site. |
| Get-BrokerUser                     | Gets broker users configured for this site.                                 |
| Group-BrokerDesktop                | Groups and counts desktops with the same value for a specified property.    |
| Group-BrokerMachine                | Groups and counts machines with the same value for a specified property.    |
| Group-BrokerSession                | Groups and counts sessions with the same value for a specified property.    |
| Import-BrokerDesktopPolicy         | Sets the site wide Citrix Group Policy settings for the site.               |
| Move-BrokerAdminFolder             | Moves a folder to another place in the hierarchy, optionally renaming it    |
| Move-BrokerApplication             | Move a published application from one admin folder to another               |
| New-BrokerAccessPolicyRule         | Creates a new rule in the site's access policy.                             |
| New-BrokerAdminFolder              | Creates a new admin folder.                                                 |
| New-BrokerAppAssignmentPolicyRule  | Creates a new application rule in the site's assignment policy.             |
| New-BrokerAppEntitlementPolicyRule | Creates a new application rule in the site's entitlement policy.            |
| New-BrokerApplication              | Creates a new published application.                                        |
| New-BrokerAssignmentPolicyRule     | Creates a new desktop rule in the site's assignment policy.                 |
| New-BrokerCatalog                  | Adds a new catalog to the site.                                             |
| New-BrokerConfigurationSlot        | Creates a new configuration slot.                                           |
| New-BrokerConfiguredFTA            | Creates a file type association with a published application.               |

| Name                                  | Description                                                                         |
|---------------------------------------|-------------------------------------------------------------------------------------|
| New-BrokerDelayedHostingPowerAction   | Causes a power action to be queued after a delay.                                   |
| New-BrokerDesktopGroup                | Create a new desktop group for managing the brokering of groups of desktops.        |
| New-BrokerEntitlementPolicyRule       | Creates a new desktop rule in the site's entitlement policy.                        |
| New-BrokerHostingPowerAction          | Creates a new action in the power action queue.                                     |
| New-BrokerHypervisorConnection        | Creates a new hypervisor connection.                                                |
| New-BrokerIcon                        | Creates a new icon.                                                                 |
| New-BrokerMachine                     | Adds a machine that can be used to run desktops and applications.                   |
| New-BrokerMachineCommand              | Creates a new command to deliver to a desktop.                                      |
| New-BrokerMachineConfiguration        | Creates a new machine configuration associated with an existing configuration slot. |
| New-BrokerPowerTimeScheme             | Creates a new power time scheme for a desktop group.                                |
| New-BrokerRebootSchedule              | Creates a new reboot schedule for a desktop group.                                  |
| New-BrokerRemotePCAccount             | Create a new RemotePCAccount.                                                       |
| New-BrokerSessionLinger               | Creates a new session linger setting for a desktop group.                           |
| New-BrokerSessionPreLaunch            | Creates a new session pre-launch setting for a desktop group.                       |
| New-BrokerTag                         | Creates a new tag.                                                                  |
| New-BrokerUser                        | Creates a new broker user object                                                    |
| Remove-BrokerAccessPolicyRule         | Deletes a rule from the site's access policy.                                       |
| Remove-BrokerAccessPolicyRuleMetadata | Deletes AccessPolicyRule Metadata from the AccessPolicyRule objects                 |
| Remove-BrokerAdminFolder              | Removes an admin folder.                                                            |

| Name                                                       | Description                                                                    |
|------------------------------------------------------------|--------------------------------------------------------------------------------|
| <a href="#">Remove-BrokerAdminFolderMetadata</a>           | Deletes AdminFolder Metadata from the AdminFolder objects                      |
| <a href="#">Remove-BrokerAppAssignmentPolicyRule</a>       | Deletes an application rule from the site's assignment policy.                 |
| <a href="#">Remove-BrokerAppEntitlementPolicyRule</a>      | Deletes an application rule from the site's entitlement policy.                |
| <a href="#">Remove-BrokerApplication</a>                   | Deletes one or more applications, or an association of an application.         |
| <a href="#">Remove-BrokerApplicationInstanceMetadata</a>   | Deletes ApplicationInstance Metadata from the ApplicationInstance objects      |
| <a href="#">Remove-BrokerApplicationMetadata</a>           | Deletes Application Metadata from the Application objects                      |
| <a href="#">Remove-BrokerAssignmentPolicyRule</a>          | Deletes a desktop rule from the site's assignment policy.                      |
| <a href="#">Remove-BrokerAssignmentPolicyRuleMetadata</a>  | Deletes AssignmentPolicyRule Metadata from the AssignmentPolicyRule objects    |
| <a href="#">Remove-BrokerCatalog</a>                       | Removes catalogs from the site.                                                |
| <a href="#">Remove-BrokerCatalogMetadata</a>               | Deletes Catalog Metadata from the Catalog objects                              |
| <a href="#">Remove-BrokerConfigurationSlot</a>             | Removes a configuration slot.                                                  |
| <a href="#">Remove-BrokerConfigurationSlotMetadata</a>     | Deletes ConfigurationSlot Metadata from the ConfigurationSlot objects          |
| <a href="#">Remove-BrokerConfiguredFTA</a>                 | Deletes one or more configured file type associations.                         |
| <a href="#">Remove-BrokerControllerMetadata</a>            | Deletes Controller Metadata from the Controller objects                        |
| <a href="#">Remove-BrokerDelayedHostingPowerAction</a>     | Cancels one or more delayed power actions.                                     |
| <a href="#">Remove-BrokerDesktopGroup</a>                  | Remove desktop groups from the system or remove them from a Remote PC catalog. |
| <a href="#">Remove-BrokerDesktopGroupMetadata</a>          | Deletes DesktopGroup Metadata from the DesktopGroup objects                    |
| <a href="#">Remove-BrokerEntitlementPolicyRule</a>         | Deletes a desktop rule from the site's entitlement policy.                     |
| <a href="#">Remove-BrokerEntitlementPolicyRuleMetadata</a> | Deletes EntitlementPolicyRule Metadata from the                                |

| Name                                                      | Description                                                                                    |
|-----------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <a href="#">Remove-BrokerHostingPowerAction</a>           | EntitlementPolicyRule objects<br>Cancel one or more pending power actions.                     |
| <a href="#">Remove-BrokerHostingPowerActionMetadata</a>   | Deletes HostingPowerAction Metadata from the HostingPowerAction objects                        |
| <a href="#">Remove-BrokerHypervisorAlertMetadata</a>      | Deletes HypervisorAlert Metadata from the HypervisorAlert objects                              |
| <a href="#">Remove-BrokerHypervisorConnection</a>         | Removes a hypervisor connection from the system.                                               |
| <a href="#">Remove-BrokerHypervisorConnectionMetadata</a> | Deletes HypervisorConnection Metadata from the HypervisorConnection objects                    |
| <a href="#">Remove-BrokerIcon</a>                         | Remove an icon.                                                                                |
| <a href="#">Remove-BrokerIconMetadata</a>                 | Deletes Icon Metadata from the Icon objects                                                    |
| <a href="#">Remove-BrokerImportedFTA</a>                  | Deletes one or more imported file type associations.                                           |
| <a href="#">Remove-BrokerLease</a>                        | Remove the specified lease in the Database.                                                    |
| <a href="#">Remove-BrokerLeaseMetadata</a>                | Deletes Lease Metadata from the Lease objects                                                  |
| <a href="#">Remove-BrokerMachine</a>                      | Removes one or more machines from its desktop group or catalog.                                |
| <a href="#">Remove-BrokerMachineCommand</a>               | Cancel a pending command queued for delivery to a desktop.                                     |
| <a href="#">Remove-BrokerMachineCommandMetadata</a>       | Deletes MachineCommand Metadata from the MachineCommand objects                                |
| <a href="#">Remove-BrokerMachineConfiguration</a>         | Deletes a machine configuration from the site or removes the association from a desktop group. |
| <a href="#">Remove-BrokerMachineConfigurationMetadata</a> | Deletes MachineConfiguration Metadata from the MachineConfiguration objects                    |
| <a href="#">Remove-BrokerMachineMetadata</a>              | Deletes Machine Metadata from the Machine objects                                              |
| <a href="#">Remove-BrokerPowerTimeScheme</a>              | Deletes an existing power time scheme.                                                         |
| <a href="#">Remove-BrokerPowerTimeSchemeMetadata</a>      | Deletes PowerTimeScheme Metadata from the PowerTimeScheme objects                              |

| <b>Remove-BrokerRebootCycleMetadata</b><br><b>Name</b> | <b>Description</b>                                                   |
|--------------------------------------------------------|----------------------------------------------------------------------|
| Remove-BrokerRebootSchedule                            | Removes the reboot schedule.                                         |
| Remove-BrokerRemotePCAccount                           | Delete RemotePCAccounts from the system.                             |
| Remove-BrokerScope                                     | Remove the specified catalog/desktop group from the given scope(s).  |
| Remove-BrokerSessionLinger                             | Removes a session linger setting.                                    |
| Remove-BrokerSessionMetadata                           | Deletes Session Metadata from the Session objects                    |
| Remove-BrokerSessionPreLaunch                          | Removes a session pre-launch setting.                                |
| Remove-BrokerSiteMetadata                              | Deletes Site Metadata from the Site objects                          |
| Remove-BrokerTag                                       | Removes a tag from the system or clears a tag to object association. |
| Remove-BrokerTagMetadata                               | Deletes Tag Metadata from the Tag objects                            |
| Remove-BrokerUser                                      | Remove broker user objects from another broker object                |
| Rename-BrokerAccessPolicyRule                          | Renames a rule in the site's access policy.                          |
| Rename-BrokerAdminFolder                               | Renames a folder                                                     |
| Rename-BrokerAppAssignmentPolicyRule                   | Renames an application rule in the site's assignment policy.         |
| Rename-BrokerAppEntitlementPolicyRule                  | Renames an application rule in the site's entitlement policy.        |
| Rename-BrokerApplication                               | Renames an application.                                              |
| Rename-BrokerAssignmentPolicyRule                      | Renames a desktop rule in the site's assignment policy.              |
| Rename-BrokerCatalog                                   | Renames a catalog.                                                   |
| Rename-BrokerDesktopGroup                              | Renames a desktop group.                                             |
| Rename-BrokerEntitlementPolicyRule                     | Renames a desktop rule in the site's entitlement policy.             |

| Name                                                   | Description                                                                                |
|--------------------------------------------------------|--------------------------------------------------------------------------------------------|
| <a href="#">Rename-BrokerMachineConfiguration</a>      | Renames a machine configuration.                                                           |
| <a href="#">Rename-BrokerPowerTimeScheme</a>           | Changes the name of an existing power time scheme.                                         |
| <a href="#">Rename-BrokerTag</a>                       | Rename one or more tags.                                                                   |
| <a href="#">Reset-BrokerLicensingConnection</a>        | Resets the broker's license server connection.                                             |
| <a href="#">Reset-BrokerServiceGroupMembership</a>     | Reloads the access permissions and configuration service locations for the Broker Service. |
| <a href="#">Send-BrokerSessionMessage</a>              | Sends a message to a session.                                                              |
| <a href="#">Set-BrokerAccessPolicyRule</a>             | Modifies an existing rule in the site's access policy.                                     |
| <a href="#">Set-BrokerAccessPolicyRuleMetadata</a>     | Creates/Updates metadata key-value pairs for AccessPolicyRule                              |
| <a href="#">Set-BrokerAdminFolderMetadata</a>          | Creates/Updates metadata key-value pairs for AdminFolder                                   |
| <a href="#">Set-BrokerAppAssignmentPolicyRule</a>      | Modifies an existing application rule in the site's assignment policy.                     |
| <a href="#">Set-BrokerAppEntitlementPolicyRule</a>     | Modifies an existing application rule in the site's entitlement policy.                    |
| <a href="#">Set-BrokerApplication</a>                  | Changes the settings of an application to the value specified in the command.              |
| <a href="#">Set-BrokerApplicationInstanceMetadata</a>  | Creates/Updates metadata key-value pairs for ApplicationInstance                           |
| <a href="#">Set-BrokerApplicationMetadata</a>          | Creates/Updates metadata key-value pairs for Application                                   |
| <a href="#">Set-BrokerAssignmentPolicyRule</a>         | Modifies an existing desktop rule in the site's assignment policy.                         |
| <a href="#">Set-BrokerAssignmentPolicyRuleMetadata</a> | Creates/Updates metadata key-value pairs for AssignmentPolicyRule                          |
| <a href="#">Set-BrokerCatalog</a>                      | Sets the properties of a catalog.                                                          |
| <a href="#">Set-BrokerCatalogMetadata</a>              | Creates/Updates metadata key-value pairs for Catalog                                       |
| <a href="#">Set-BrokerConfigurationSlotMetadata</a>    | Creates/Updates metadata key-value pairs for ConfigurationSlot                             |

| Name                                                    | Description                                                         |
|---------------------------------------------------------|---------------------------------------------------------------------|
| <a href="#">Set-BrokerControllerMetadata</a>            | Creates/Updates metadata key-value pairs for Controller             |
| <a href="#">Set-BrokerDBConnection</a>                  | Configures a database connection for the Broker Service.            |
| <a href="#">Set-BrokerDesktopGroup</a>                  | Adjusts the settings of a broker desktop group.                     |
| <a href="#">Set-BrokerDesktopGroupMetadata</a>          | Creates/Updates metadata key-value pairs for DesktopGroup           |
| <a href="#">Set-BrokerEntitlementPolicyRule</a>         | Modifies an existing desktop rule in the site's entitlement policy. |
| <a href="#">Set-BrokerEntitlementPolicyRuleMetadata</a> | Creates/Updates metadata key-value pairs for EntitlementPolicyRule  |
| <a href="#">Set-BrokerHostingPowerAction</a>            | Changes the priority of one or more pending power actions.          |
| <a href="#">Set-BrokerHostingPowerActionMetadata</a>    | Creates/Updates metadata key-value pairs for HostingPowerAction     |
| <a href="#">Set-BrokerHypervisorAlertMetadata</a>       | Creates/Updates metadata key-value pairs for HypervisorAlert        |
| <a href="#">Set-BrokerHypervisorConnection</a>          | Sets the properties of a hypervisor connection.                     |
| <a href="#">Set-BrokerHypervisorConnectionMetadata</a>  | Creates/Updates metadata key-value pairs for HypervisorConnection   |
| <a href="#">Set-BrokerIconMetadata</a>                  | Creates/Updates metadata key-value pairs for Icon                   |
| <a href="#">Set-BrokerLeaseMetadata</a>                 | Creates/Updates metadata key-value pairs for Lease                  |
| <a href="#">Set-BrokerMachine</a>                       | Sets properties on a machine.                                       |
| <a href="#">Set-BrokerMachineCatalog</a>                | Moves one or more machines into a different catalog.                |
| <a href="#">Set-BrokerMachineCommandMetadata</a>        | Creates/Updates metadata key-value pairs for MachineCommand         |
| <a href="#">Set-BrokerMachineConfiguration</a>          | Sets the properties of a machine configuration.                     |
| <a href="#">Set-BrokerMachineConfigurationMetadata</a>  | Creates/Updates metadata key-value pairs for MachineConfiguration   |
| <a href="#">Set-BrokerMachineMaintenanceMode</a>        | Sets whether the specified machine(s) are in maintenance mode.      |

| Name                                               | Description                                                                                     |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <a href="#">Set-BrokerMachineMetadata</a>          | Creates/Updates metadata key-value pairs for Machine                                            |
| <a href="#">Set-BrokerPowerTimeScheme</a>          | Modifies an existing power time scheme.                                                         |
| <a href="#">Set-BrokerPowerTimeSchemeMetadata</a>  | Creates/Updates metadata key-value pairs for PowerTimeScheme                                    |
| <a href="#">Set-BrokerPrivateDesktop</a>           | Change the settings of a private desktop.                                                       |
| <a href="#">Set-BrokerRebootCycleMetadata</a>      | Creates/Updates metadata key-value pairs for RebootCycle                                        |
| <a href="#">Set-BrokerRebootSchedule</a>           | Updates the values of one or more desktop group reboot schedules.                               |
| <a href="#">Set-BrokerRemotePCAccount</a>          | Modify one or more RemotePCAccounts.                                                            |
| <a href="#">Set-BrokerSession</a>                  | Sets properties of a session.                                                                   |
| <a href="#">Set-BrokerSessionLinger</a>            | Updates the values of one or more desktop group session linger settings.                        |
| <a href="#">Set-BrokerSessionMetadata</a>          | Creates/Updates metadata key-value pairs for Session                                            |
| <a href="#">Set-BrokerSessionPreLaunch</a>         | Updates the values of one or more desktop group session pre-launch settings.                    |
| <a href="#">Set-BrokerSharedDesktop</a>            | Change the settings of a shared desktop.                                                        |
| <a href="#">Set-BrokerSite</a>                     | Changes the overall settings of the current XenDesktop broker site.                             |
| <a href="#">Set-BrokerSiteMetadata</a>             | Creates/Updates metadata key-value pairs for Site                                               |
| <a href="#">Set-BrokerTagMetadata</a>              | Creates/Updates metadata key-value pairs for Tag                                                |
| <a href="#">Start-BrokerCatalogPvdImagePrepare</a> | Start the PVD Image prepare process in the Broker for the machines in the specified catalog(s). |
| <a href="#">Start-BrokerMachinePvdImagePrepare</a> | Start the PVD Image prepare process in the Broker for the specified machine(s).                 |
| <a href="#">Start-BrokerNaturalRebootCycle</a>     | Reboots all machines from the specified catalog when they are not in use.                       |



| Name                                             | Description                                                                                                 |
|--------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| Start-BrokerRebootCycle                          | Creates and starts a reboot cycle for each desktop group that contains machines from the specified catalog. |
| Stop-BrokerRebootCycle                           | Cancels the specified reboot cycle.                                                                         |
| Stop-BrokerSession                               | Stop or log off a session.                                                                                  |
| Test-BrokerAccessPolicyRuleNameAvailable         | Determine whether the proposed AccessPolicyRule Name is available for use.                                  |
| Test-BrokerAppAssignmentPolicyRuleNameAvailable  | Determine whether the proposed AppAssignmentPolicyRule Name is available for use.                           |
| Test-BrokerAppEntitlementPolicyRuleNameAvailable | Determine whether the proposed AppEntitlementPolicyRule Name is available for use.                          |
| Test-BrokerApplicationNameAvailable              | Determine whether the proposed Application Name is available for use.                                       |
| Test-BrokerAssignmentPolicyRuleNameAvailable     | Determine whether the proposed AssignmentPolicyRule Name is available for use.                              |
| Test-BrokerCatalogNameAvailable                  | Determine whether the proposed Catalog Name is available for use.                                           |
| Test-BrokerDBConnection                          | Tests whether a database is suitable for use by the Citrix Broker Service.                                  |
| Test-BrokerDesktopGroupNameAvailable             | Determine whether the proposed DesktopGroup Name is available for use.                                      |
| Test-BrokerEntitlementPolicyRuleNameAvailable    | Determine whether the proposed EntitlementPolicyRule Name is available for use.                             |
| Test-BrokerLicenseServer                         | Tests whether or not a license server can be used by the broker.                                            |
| Test-BrokerMachineNameAvailable                  | Determine whether the proposed Machine MachineName is available for use.                                    |
| Test-BrokerPowerTimeSchemeNameAvailable          | Determine whether the proposed PowerTimeScheme Name is available for use.                                   |
| Test-BrokerRemotePCAccountNameAvailable          | Determine whether the proposed RemotePCAccount OU is available for use.                                     |
| Update-BrokerImportedFTA                         | Imports or updates all of the file type associations for the specified worker.                              |
| Update-BrokerLocalLeaseCache                     | Flushes the local lease cache.                                                                              |

| Name              | Description                                                   |
|-------------------|---------------------------------------------------------------|
| e-BrokerNameCache | Administrative operations on the user and machine name cache. |

# about\_Broker\_AccessPolicy

Jan 04, 2017

## TOPIC

Citrix Broker SDK - Access Policy

## SHORT DESCRIPTION

Controls client-connection-based access to desktop groups.

## LONG DESCRIPTION

The site's access policy defines rules controlling a user's access to desktop groups. Access checks are based on details of the user's connection from their user device to the site. Think of the access policy informally as a connection-based firewall.

The access policy comprises a set of rules. Each rule:

- o Relates to a single desktop group.
- o Contains a set of connection filters and access right controls.

Multiple rules can apply to the same desktop group.

By default, users have no access to any desktop group within a site. A user gains access to a group when their connection's details match the connection filters of one or more rules in the access policy.

The access policy also grants control rights over desktop and application sessions. For example, it can specify which protocols are allowed for a connection from a given endpoint, and whether the user can restart their machine.

To use a resource published by a site, the user must have both access to the desktop group that contains it, and an entitlement to use the resource. Entitlements are typically granted by the site entitlement and assignment policies; see [about\\_Broker\\_Policies](#) for more information.

## ACCESS POLICY RULES

A single access policy rule relates to a single specified desktop group and comprises a set of connection filters and access right grants as described below.

Each rule can be individually enabled or disabled. A disabled rule is ignored when the access policy is evaluated.

## CONNECTION FILTERS OVERVIEW

The connection filters in an access policy rule comprise the following:

- o Local/remote client (SmartAccess) filters
- o Client IP address filters
- o Client name filters
- o User filters

All filters have an include and exclude form that can be individually enabled or disabled. For a rule to be considered when the access policy is evaluated, at least one connection include filter must be enabled. By default, all filters, both include and exclude, are disabled.

The detailed behavior of connection filters is covered later.

## ACCESS RIGHT CONTROLS OVERVIEW

The access right controls in an access policy rule comprise the following:

- o Allowed protocols
- o Whether machine restart, or programmatic session logoff, is allowed

The detailed behavior of access right controls is covered later.

## DETAILS OF CONNECTION FILTERS

To gain access to a desktop group the user's connection must match the filter criteria of at least one access policy rule for that group.

To match a rule, a connection must match all the rule's enabled include connection filters and must not match any of the rule's enabled exclude filters. That is, entries in exclude filters take priority.

Because all rules are evaluated independently, if an exclude filter match prevents a connection gaining access to a desktop group through one rule, the connection may still gain access to the same group through a different rule.

The filters are described in pairs below, but within a single rule a match against any exclude filter prevents a connection from gaining access through that rule irrespective of which include filters within the rule were also matched.

## SMARTACCESS FILTERS

SmartAccess filters allow filtering based on whether the client is directly connected (for example over a local area network (LAN)) or through Access Gateway. For connections through Access Gateway further filtering can be performed based on the tags supplied from Access Gateway itself. The key properties of SmartAccess filters are:

- o AllowedConnections (include filter: Filtered, NotViaAG, ViaAG)

This property controls the behavior of the include filter. The default value is Filtered. The possible values are as follows:

-- Filtered (default)

The filter matches any user connection not through Access Gateway. In addition, the filter may match user connections through Access Gateway subject to the following: if the IncludedSmartAccessTags property is empty, any such connection matches. However, if the property is not empty at least one SmartAccess tag from the filter property must match a SmartAccess tag supplied with the user's

connection.

#### -- NotViaAG

The filter matches only user connections not through Access Gateway. The contents of the IncludedSmartAccessTags property are ignored.

#### -- ViaAG

The filter matches only user connections through Access Gateway. If the IncludedSmartAccessTags property is empty, any such connection matches. However, if the property is not empty at least one SmartAccess tag from the filter property must match a SmartAccess tag supplied with the user's connection.

The IncludedSmartAccessTags property referred to above forms part of the include filter and is used if AllowedConnections is set to Filtered or ViaAG. It comprises a simple list of Access Gateway tags that are matched against those provided in the user's connection details.

#### o ExcludedSmartAccessTags (exclude filter)

A simple list of Access Gateway tags that are matched against those provided in the user's connection details. If any tag in the list matches one supplied with the user's connection, the user's connection does not match the access policy rule containing the filter.

The exclude filter has no setting corresponding to the AllowedConnections property so its behavior is determined only by the ExcludedSmartAccessTags property.

SmartAccess filters are typically used to control local (through a LAN) and remote (through Access Gateway) access to a site. A common model is to define two access policy rules for a group, one for local access and one for remote. The remote rule might impose restrictions on the user device having appropriate antivirus software installed, and potentially exclude certain user groups who would be allowed access over the corporate LAN (see USER FILTERS below).

## CLIENT IP FILTERS

Client IP filters allow filtering based on the IP address of the user's device. The key properties of client IP filters are:

#### o IncludedClientIPs (include filter)

A simple list of numeric IP address ranges that are matched against the user device. The filter matches if the device address falls within any of the ranges in the list.

- o ExcludedClientIPs (exclude filter)

A simple list of numeric IP address ranges that are matched against the user device. If any entry matches the device address, the user's connection does not match the access policy rule containing the filter.

An IP address range in these filters can be specified as a simple IP address or as a range using a conventional subnet mask.

## CLIENT NAME FILTERS

Client name filters allow filtering based on the name of the user's device. The key properties of client name filters are:

- o IncludedClientNames (include filter)

A simple list of names that are matched against the user device. The filter matches if the device name matches any value in the list.

- o ExcludedClientNames (exclude filter)

A simple list of device names that are matched against the user device. If any entry matches the device name, the user's connection does not match the access policy rule containing the filter.

Note: The form of the device name presented to the site depends on the site configuration. For example, by default in these filters you cannot use the form of the name presented by Web Interface.

## USER FILTERS

User filters allow filtering based on the identity of the user. The key properties of user filters are:

- o AllowedUsers (include filter: Filtered, AnyAuthenticated, Any)

This property controls the behavior of the include filter. The default value is Filtered. The possible values are as follows:

- Filtered (default)

The filter matches if the user's logon token contains one or more users or user groups matching those specified in the IncludedUsers property. The IncludedUsers property is a simple list of users or user groups and is used only when the AllowedUsers property is set to Filtered.

- AnyAuthenticated

The filter matches any authenticated Microsoft Windows user. The

contents of the IncludedUsers property are ignored.

-- Any

The filter matches any user. The contents of the IncludedUsers property are ignored. In the current implementation this value is handled in the same way as AnyAuthenticated.

- o ExcludedUsers (exclude filter)

A simple list of users or user groups. If any entry matches one in the user's logon token, the user's connection does not match the access policy rule containing the filter.

The exclude filter has no setting corresponding to the AllowedUsers property so its behavior is determined only by the ExcludedUsers property.

## DETAILS OF ACCESS RIGHT CONTROLS

The access right controls of an access policy rule determine rights that the user has over any desktop or application session that they obtain from the rule's desktop group.

The rights apply only if the user's connection matches the connection filters of a rule, and only if the user also has an entitlement to a desktop or application session from the associated desktop group.

The following properties define the access rights:

- o AllowedProtocols

A simple list of communication protocols over which connections can be made to resources published by the desktop group. For example, use this to restrict protocols with high bandwidth requirements to connections originating from a LAN.

- o AllowRestart

For single-session power-managed machines, allows the user to restart the machine (the machine is powered off using the capabilities of its hypervisor). For multi-session machines the user's session is simply logged off.

For a given connection, if multiple rules result in access being granted to a session from a desktop group, the user's rights are the combined rights of all the rules that matched for that group. The allowed protocol lists from all the rules are combined, and the user is granted restart rights if any one rule has AllowRestart set.

SEE ALSO

about\_Broker\_Policies  
about\_Broker\_AssignmentPolicy  
about\_Broker\_EntitlementPolicy  
New-BrokerAccessPolicyRule  
Get-BrokerAccessPolicyRule  
Set-BrokerAccessPolicyRule  
Rename-BrokerAccessPolicyRule  
Remove-BrokerAccessPolicyRule  
New-BrokerAssignmentPolicyRule  
New-BrokerEntitlementPolicyRule  
New-BrokerAppAssignmentPolicyRule  
New-BrokerAppEntitlementPolicyRule  
Add-BrokerUser



# about\_Broker\_Applications

Jan 04, 2017

## TOPIC

Citrix Broker SDK - Applications

## SHORT DESCRIPTION

Describes how to publish and manage hosted applications.

## LONG DESCRIPTION

Published applications allow your users to launch applications as if they were installed on their devices when in fact they are hosted remotely. The applications are normally launched in a seamless window, meaning that users see only the application window and not an additional desktop.

Published applications are hosted on either desktop operating systems or server operating systems. Applications are published to users and desktop groups. As such, conceptually they exist "on top" of desktop groups, which are themselves built on top of catalogs. See [about\\_Broker\\_Concepts](#) for more information on catalogs.

There are two types of applications:

- o HostedOnDesktop - application runs on a remote machine and is displayed on the local client desktop.
- o InstalledOnClient - application is installed and run on the local client machine and has its window overlaid on to a remote desktop.

## HOSTING APPLICATIONS

There are three main ways of hosting an application: using a private single-session VDI desktop, a shared single-session VDI desktop, and on shared multi-session server operating systems.

An application hosted on a shared single-session VDI desktop, when launched, is hosted on a random machine within the desktop group. An application hosted on a private single-session VDI desktop ensures that when a user launches the application it is always hosted on the same machine. An application hosted on shared multi-session server operating systems ensures that when a user launches the application it is always hosted on one of the least loaded servers in the desktop group.

You control how the application is hosted based on the kind of desktop group that you choose. Shared desktop groups randomly select a machine, and these desktop groups can only be created from catalogs with a `Random AllocationType`. On the other hand, private desktop groups host the application on the same machine for that user every time, and these desktop group types can only be created from catalogs with a `Permanent AllocationType`.

## USER ACCESS & ASSIGNMENT

Users are not assigned an application directly, but instead they are required to first have access to the desktop groups on which the applications are published through access policy rules.

With shared desktop groups, access to a published application also needs an application entitlement policy rule. An application entitlement policy rule defines the set of users who are allowed per-session access to desktops in a specified desktop group.

With private desktop groups, access to a published application also needs an application assignment policy rule. An application assignment policy rule defines the set of users who are allowed access to a single application session in a desktop group.

Users are assigned private machines in one of two ways: pre-assigned or assign-on-first-use. Pre-assigned means that individual user accounts have been specified for the machines within that desktop group. A single machine can only have a single user account (not a group account) assigned to it, and likewise a user can only be assigned a single machine within a desktop group.

Assign-on-first-use requires less configuration. Machines are assigned to users the first time they log on. Rather than allocating users directly to machines, instead users are assigned to the private desktop group, either through individual user accounts or through user group accounts. Then, when a user assigned to the desktop group logs on, a machine is automatically allocated to them.

## USER VISIBILITY

Users can be further filtered by configuring the visibility filter on top of the application. This restricts the application to a subset of the users that were granted access by the access policy and entitlement/assignment policy rules on the desktop group.

## MULTIPLE DESKTOP GROUPS

You can publish an application to multiple desktop groups, which have to be of the same kind. Generally, an application that is published only to private desktop groups is referred to as a "private application". An application that is only published to shared desktop groups is referred to as a "shared application".

## LICENSING

Hosted applications need the appropriate licenses to exist on the license server to be functional.

## NAMING

Applications have three names that identify them: the Name, BrowserName and the PublishedName. The Name is unique for each application, is not visible to the user, and is primarily used for administrative purpose. The BrowserName is unique across the entire site, and is primarily used internally. The PublishedName is not unique and is the name seen by end users who have access to this application.

When creating an application, you only need to specify the Name. If no BrowserName is specified one is automatically generated. If no PublishedName is specified by default it is the same as the Name.

The following special characters are not allowed in the Name, BrowserName or the PublishedName properties: \ / ; # . \* ? = &lt; &gt; | [ ] ( ) ' " ' "

In addition the ` character is not allowed in the Name property.

To change the PublishedName or BrowserName of an application you must use the Set-BrokerApplication cmdlet.

To change the Name of an application you must use the Rename-BrokerApplication cmdlet.

## OPTIONAL

You can configure file-type associations for applications, so that if a user double-clicks a document icon on their device, a published application automatically starts. For more information, see the help for `New-BrokerConfiguredFTA`.

You can apply tags to applications as another convenient way to further organize (and search for) applications. For more information, see the help for `New-BrokerTag`.

## CMDLETS

### New-BrokerApplication

Creates an application for publishing after the needed desktop groups, access policy and entitlement/assignment policy rules have been created.

### Add-BrokerApplication

Adds one or more applications to a desktop group.

### Get-BrokerApplication

Gets one or more applications.

### Remove-BrokerApplication

Deletes one or more applications or it can be used to delete just the association of an application to a desktop group.

### Rename-BrokerApplication

Changes the Name of an application.

### Set-BrokerApplication

Changes the settings of application objects, except their names.

## EXAMPLES

## SHARED APPLICATION (SingleSession)

You have created a catalog with a Random AllocationType and SingleSession SessionSupport. It contains two machines called worker1 and worker2, both in the ACME domain. You publish an application with shared hosting as follows:

```
C:\PS> Write-Host "Create a desktop group, and add machines to it"
C:\PS> $dg = New-BrokerDesktopGroup "Shared Application Group"
-DesktopKind Shared -DeliveryType AppsOnly -SessionSupport 'SingleSession'
C:\PS> $m1 = Get-BrokerMachine -MachineName "ACMEworker1"
C:\PS> $m2 = Get-BrokerMachine -MachineName "ACMEworker2"
C:\PS> Add-BrokerMachine $m1 -DesktopGroup $dg
C:\PS> Add-BrokerMachine $m2 -DesktopGroup $dg
C:\PS> Write-Host "Create access policy rule for desktop group"
C:\PS> New-BrokerAccessPolicyRule -Name "Shared Application Group - Allow Everyone Access"
-Enabled $true -DesktopGroupUid $dg.Uid -IncludedUserFilterEnabled $true
-AllowedProtocols @("HDX") -AllowedUsers AnyAuthenticated
C:\PS> Write-Host "Create application entitlement policy for desktop group"
C:\PS> New-BrokerAppEntitlementPolicyRule -Name "Shared Application Group - App Entitlement"
-IncludedUsers 'ACME\Domain Users' -DesktopGroupUid $dg.Uid -Enabled $true
C:\PS> Write-Host "Create an application"
C:\PS> New-BrokerApplication -Name "Notepad" -PublishedName "Notepad"
-CommandLineExecutable "notepad.exe" -DesktopGroup $dg.Uid
```

In turn, this set of commands: creates a shared single session desktop group for applications delivery; adds two machines (from a catalog with a Random AllocationType and SingleSession SessionSupport) to the desktop group; creates access policy and application entitlement policy rules; creates an application; specifies its name, the executable, and links the application to the desktop group that will host it.

## SHARED APPLICATION (MultiSession)

You have created a catalog with a Random AllocationType and MultiSession SessionSupport. It contains one machine called worker1 in the ACME domain. You publish an application with shared hosting as follows:

```
C:\PS> Write-Host "Create a desktop group, and add machines to it"
C:\PS> $dg = New-BrokerDesktopGroup "Shared Application Group"
-DesktopKind Shared -DeliveryType AppsOnly -SessionSupport 'MultiSession'
C:\PS> $m1 = Get-BrokerMachine -MachineName "ACMEworker1"
C:\PS> Add-BrokerMachine $m1 -DesktopGroup $dg
C:\PS> Write-Host "Create access policy rule for desktop group"
C:\PS> New-BrokerAccessPolicyRule -Name "Shared Application Group - Allow Everyone Access"
-Enabled $true -DesktopGroupUid $dg.Uid -IncludedUserFilterEnabled $true
-AllowedProtocols @("HDX") -AllowedUsers AnyAuthenticated
C:\PS> Write-Host "Create application entitlement policy for desktop group"
C:\PS> New-BrokerAppEntitlementPolicyRule -Name "Shared Application Group - App Entitlement"
-IncludedUsers 'ACME\Domain Users' -DesktopGroupUid $dg.Uid -Enabled $true
C:\PS> Write-Host "Create an application"
```

```
C:\PS> New-BrokerApplication -Name "Notepad" -PublishedName "Notepad"  
-CommandLineExecutable "notepad.exe" -DesktopGroup $dg.Uid
```

In turn, this set of commands: creates a shared multi session desktop group for applications delivery; adds one machine (from a catalog with a Random AllocationType and MultiSession SessionSupport) to the desktop group; creates access policy and application entitlement policy rules; creates an application; specifies its name, the executable, and links the application to the desktop group that will host it.

#### PRIVATE PRE-ASSIGNED APPLICATION

You have a catalog with a Permanent AllocationType and SingleSession SessionSupport. It contains two machines called worker1 and worker2, both in the ACME domain. You publish an application with private hosting using pre-assigned machines as follows:

```
C:\PS> Write-Host "Create a desktop group, and add machines to it"  
C:\PS> $dg = New-BrokerDesktopGroup "Private App G1" -DesktopKind Private  
-DeliveryType AppsOnly -SessionSupport 'SingleSession'  
C:\PS> $m1 = Get-BrokerMachine -MachineName "ACME\worker1"  
C:\PS> $m2 = Get-BrokerMachine -MachineName "ACME\worker2"  
C:\PS> Add-BrokerMachine $m1 -DesktopGroup $dg  
C:\PS> Add-BrokerMachine $m2 -DesktopGroup $dg  
C:\PS> Write-Host "Setting access policy rule for desktop group"  
C:\PS> New-BrokerAccessPolicyRule -Name "Private App G1 - Allow Everyone Access"  
-Enabled $true -DesktopGroupUid $dg.Uid -IncludedUserFilterEnabled $true  
-AllowedProtocols @("HDX") -AllowedUsers AnyAuthenticated  
C:\PS> Write-Host "Pre-Assign users to the machines"  
C:\PS> Add-BrokerUser "ACME\user1" -Machine $m1  
C:\PS> Add-BrokerUser "ACME\user2" -Machine $m2  
C:\PS> Write-Host "Create an application"  
C:\PS> New-BrokerApplication -Name "Notepad" -PublishedName "Notepad"  
-CommandLineExecutable "notepad.exe" -DesktopGroup $dg.Uid
```

In turn, this set of commands: creates a private desktop group for applications delivery; adds two machines (from a catalog with a Permanent AllocationType and SingleSession SessionSupport) to the desktop group; creates access policy; creates two user objects; pre-assigns a user to each machine; creates the application; assigns users to it; and links the application to the desktop group that will host it.

#### PRIVATE, ASSIGN-ON-FIRST-USE APPLICATION

You have a catalog created with a Permanent AllocationType and SingleSession SessionSupport. It contains two machines called worker1 and worker2, both in the ACME domain. You publish an application with private hosting using assign-on-first-use machines as follows:

```
C:\PS> Write-Host "Create a desktop group, and add machines to it"
```

```
C:\PS> $dg = New-BrokerDesktopGroup "Private App G2" -DesktopKind Private
-DeliveryType AppsOnly -SessionSupport 'SingleSession'
C:\PS> $m1 = Get-BrokerMachine -MachineName "ACME\worker1"
C:\PS> $m2 = Get-BrokerMachine -MachineName "ACME\worker2"
C:\PS> Add-BrokerMachine $m1 -DesktopGroup $dg
C:\PS> Add-BrokerMachine $m2 -DesktopGroup $dg
C:\PS> Write-Host "Setting access policy rule for desktop group"
C:\PS> New-BrokerAccessPolicyRule -Name "Private App G1 - Allow Everyone Access"
-Enabled $true -DesktopGroupUid $dg.Uid -IncludedUserFilterEnabled $true
-AllowedProtocols @("HDX") -AllowedUsers AnyAuthenticated
C:\PS> Write-Host "Create application assignment policy for desktop group"
C:\PS> New-BrokerAppAssignmentPolicyRule -Name "Private App G2 - App Assignment"
-IncludedUsers 'ACME\Domain Users' -DesktopGroupUid $dg.Uid -Enabled $true
C:\PS> Write-Host "Create an application"
C:\PS> New-BrokerApplication -Name "Notepad" -PublishedName "Notepad"
-CommandLineExecutable "notepad.exe" -DesktopGroup $dg.Uid
```

In turn, this set of commands: creates a private single session desktop group for applications delivery; adds two machines (from a catalog with a Permanent AllocationType and SingleSession SessionSupport) to the desktop group; creates access policy and application assignment policy rules; creates the application, and links the application to the desktop group that will host it.

## SEE ALSO

[about\\_Broker\\_Concepts](#)

[about\\_Broker\\_Desktops](#)

[New-BrokerApplication](#)

[Add-BrokerApplication](#)

[Remove-BrokerApplication](#)

[Rename-BrokerApplication](#)

[Set-BrokerApplication](#)

[New-BrokerAppAssignmentPolicyRule](#)

[Remove-BrokerAppAssignmentPolicyRule](#)

[Set-BrokerAppAssignmentPolicyRule](#)

[New-BrokerAppEntitlementPolicyRule](#)

[Remove-BrokerAppEntitlementPolicyRule](#)

[Set-BrokerAppEntitlementPolicyRule](#)



# about\_Broker\_AssignmentPolicy

Jan 04, 2017

## TOPIC

Citrix Broker SDK - Assignment Policy

## SHORT DESCRIPTION

Controls the automatic, permanent assignment of machines to users.

## LONG DESCRIPTION

The site's assignment policy defines rules controlling automatic assignment of machines to users in a process referred to as Assign On First Use (AOFU).

In this assignment model, a desktop group is initially populated with machines that have no assignments, and users are granted entitlements to obtain a machine selected at random from the pool by self-service assignment. Once made, the assignment is permanent. The resulting assigned machines can be used to deliver either desktop or application sessions depending on the delivery type of the desktop group.

The assignment policy comprises a set of rules. The policy can be applied only to desktop groups of desktop kind Private.

For assignment of machines to deliver desktop sessions:

- o Multiple rules can apply to the same desktop group.
- o Each rule grants an entitlement to one or more machines.

For assignment of machines to deliver application sessions:

- o Only a single rule can apply to a given desktop group.
- o Each rule grants an entitlement to a single machine.
- o Although only a single application assignment rule can be defined for a group, a user can still launch multiple applications from that group because the applications all run within the same session on the assigned machine.

Once a machine is assigned to a user by an assignment policy rule, the rule plays no further part in controlling access to that machine. The rule can be changed, or even removed, without impacting the user's access to the machine.

Rules for desktop and application machine assignments are distinct. Desktop assignments are managed through the BrokerAssignmentPolicyRule SDK object, and application rules through the BrokerAppAssignmentPolicyRule object.

Desktop machine assignment rules can be created only for desktop groups with delivery type DesktopsOnly, whereas an application machine assignment rule can be created only for delivery type AppsOnly.

Because desktop groups of assigned machines do not allow delivery type DesktopsAndApps, desktop machine assignment



and application machine assignment rules cannot exist for the same desktop group.

Assignment policy rules are also used to configure the RemotePC feature where their detailed usage differs. For more information on the RemotePC feature see "help about\_Broker\_RemotePC".

For an entitlement granted by the assignment policy to be available to a user, the site's access policy must also grant them access to the desktop group.

## ASSIGNMENT POLICY RULES

Each assignment policy rule has the following key properties:

- o The desktop group to which it applies (one rule only ever applies to one group).
- o The users to which machines can be assigned by the rule.

Additionally for desktop assignment rules, the following properties exist:

- o The published name of the entitlement (visible to the user).
- o The number of machines (entitlements) to which the rule grants access.
- o The properties that a newly assigned desktop acquires.

If multiple desktop assignment rules entitle a user to machines from the same desktop group, their total machine entitlement is the sum of those granted by all those rules. The properties of the desktop sessions obtained may differ depending on which of the entitlements the user selects to start a session.

Each rule can be individually enabled or disabled. A disabled rule is ignored when the assignment policy is evaluated.

## USER FILTERS (FULL)

Each rule has two user filters, an include filter and an exclude filter:

- o The include filter contains users and user groups that are granted entitlements to machines.
- o The exclude filter contains users and user groups that are denied entitlements to machines.

If the include filter of a rule contains multiple instances of a user (either explicit or implicit), this does not alter the number of machine entitlements granted to them by the rule.

Entries in the exclude filter take priority, so if a user appears explicitly or implicitly in both filters, access is denied. Typically, you use this filter to exclude a user or group of users who would otherwise gain access because they are members of a user group specified in the include filter.

Because all rules are independently evaluated, the exclude filter can exclude only users who would otherwise gain an entitlement through the same rule's include filter. That is, if a user is in a rule's include filter but not its exclude filter, the rule

is guaranteed to grant that user access to a machine irrespective of whether the user appears in the exclude filter of other rules.

If a filter contains a user group that contains other users and groups, the filter implicitly includes all of those users and groups.

By default the exclude filter is disabled.

To maintain assignment policy rules that can be fully displayed and edited with Citrix Studio, use the simplified user filter model below and do not use the exclude filter.

## USER FILTERS (SIMPLIFIED)

The included user filter described above also supports a simplified usage model where the filter itself is disabled. When this is done, any user who has access to the desktop group through the access policy is implicitly granted an entitlement to a machine through the entitlement policy rule without the need to list the user in the rule's include filter.

This is useful in cases where the access policy for the desktop group already explicitly specifies the users who should have access.

Even if the include filter is disabled, the exclude filter can still be used to deny the entitlement to users who would otherwise gain access through the access policy alone.

This simplified usage cannot be used for RemotePC desktop groups.

## REMOTE PC USAGE

When a desktop group is configured as a RemotePC group, the usage of the assignment policy differs in the following ways:

- o Only a single rule can apply to a given desktop group.
- o The delivery type of the desktop group must be DesktopsOnly.
- o The simplified user filter model cannot be used. Users or user groups must appear explicitly in the included user filter of the assignment policy rule.
- o If the included user filter is disabled then RemotePC assignment for the group is also effectively disabled.

For more information on the RemotePC feature see "help about\_Broker\_RemotePC".

## ADDITIONAL DESKTOP ASSIGNMENT RULE PROPERTIES

Desktop assignment rules specify the following additional properties:

- o PublishedName
- o Description
- o IconUid
- o ColorDepth
- o SecureIcaRequired

The published name, description, and icon UID properties apply to the desktop entitlement itself and determine the way in which the entitlement is presented to the user in, for example, StoreFront.

The color depth and secure ICA properties apply to the desktop session that is obtained when the entitlement is used.

In all cases, these properties can be explicitly specified. However, a null value (the default) means that the corresponding property is taken from the desktop group to which the rule applies. This inheritance from groups is dynamic; if the property of the group changes, the property of the entitlement changes too.

For assignment rules these properties are copied to the newly assigned desktop when the granted entitlement is first used. If the properties on the rule are subsequently changed the properties on the user's assigned desktop do not change. The dynamic inheritance of desktop properties from the desktop group continues after the assignment has occurred if the original rule did not provide explicit property values.

## CALCULATING OVERALL MACHINE ENTITLEMENTS

Assignment rules are modified only by the SDK; they are not modified when used by the system to make automatic assignments. The number of machine entitlements offered to the user is determined by the number of machines (within the desktop group) already assigned to them and by which rules those assignments were made.

For each desktop group, the number of desktop entitlements available to the user is determined as follows:

1. The total number of entitlements for the user to machines in the group,

from all rules, is calculated.

2. The total number of assigned machines (from any source) that the user

already has in the group is determined.

3. The outstanding entitlement value is derived as the difference of the

above two numbers.

4. If the outstanding entitlement value is zero or fewer, no further

entitlements are allowed.

Otherwise, for each applicable rule, the number of entitlements is that defined by the rule itself, minus the number of desktops already assigned to the user by that rule, and capped by the outstanding entitlement value.

Desktop and application machine entitlements both follow the above rules. However, because only a single application assignment rule granting a single entitlement per group can exist, these rules are seldom a consideration for applications.

## EXAMPLES

### Simple case:

A user is entitled to a single machine in a group by a single assignment rule:

1. On first use, the user sees a single desktop entitlement.  
If the user selects this, a new desktop is assigned.
2. On subsequent uses, the user sees the assigned desktop only.  
No other entitlements are displayed.

### Complex case:

A user is entitled to two machines, one from each of two different rules, A and B, both applying to the same desktop group. In addition, the user has a machine explicitly assigned to them by the administrator within that group:

1. On first use, the user sees the administrator-assigned desktop, a single entitlement to a desktop of type A, and a single entitlement to a desktop of type B. If either of the entitlements is selected, a desktop of the appropriate type, A or B, is assigned.
2. On subsequent uses, the user sees the administrator-assigned desktop and the desktop assigned by the selected entitlement.  
No further entitlements are displayed.

## MACHINE ENTITLEMENT PRESENTATION

For desktop machine assignment rules, the user interface determines whether users can visually distinguish between an entitlement to a machine and an actual assigned desktop. This cannot be controlled by Broker SDK cmdlets.

Although the number of entitlements seen by users takes account of all of their currently assigned desktops, the dynamic state of the system can affect the user interaction. This means that entitlements are displayed even where the pool of machines is empty, or the remaining desktops are in maintenance mode or otherwise unavailable. If the user attempts to use an entitlement in these cases, they receive a no-available-desktop error.

For application machine entitlement rules, the entitlements themselves are not presented to the end user; only the applications available to the user are presented. The use of available assignments, or of already assigned machines, is managed automatically.

## NOTES

If an assignment rule grants entitlements to a user group:

- o The machine is assigned to the user who selects the entitlement, thus an assignment policy rule cannot be used to assign a machine to a user group. However, an administrator can assign a machine to a user group using the Add-BrokerUser cmdlet.
- o The number of machine entitlements specified in a desktop assignment rule applies to each member of the user group. For example, if a rule grants a user group access to two machines, every user in the group is entitled to two desktops.

The total number of machine entitlements defined by the assignment policy for a given desktop group may exceed the number of machines present in the group. A user attempting to use an entitlement when the pool of machines is empty receives a no-desktop-available error. (See CALCULATING OVERALL MACHINE ENTITLEMENTS above).

A machine assigned to a user as a result of the assignment policy remains permanently assigned unless the machine assignment itself is removed by the administrator. Such assignments are permanent even if the assignment rule is deleted, when the machine is treated as administrator-assigned.

The assignment policy cannot be used to assign a machine to more than one user. This is only possible using the Add-BrokerUser cmdlet.

#### SEE ALSO

[about\\_Broker\\_Policies](#)

[about\\_Broker\\_AccessPolicy](#)

[about\\_Broker\\_EntitlementPolicy](#)

[about\\_Broker\\_Applications](#)

[New-BrokerAssignmentPolicyRule](#)

[Get-BrokerAssignmentPolicyRule](#)

[Set-BrokerAssignmentPolicyRule](#)

[Rename-BrokerAssignmentPolicyRule](#)

[Remove-BrokerAssignmentPolicyRule](#)

[New-BrokerAppAssignmentPolicyRule](#)

[Get-BrokerAppAssignmentPolicyRule](#)

[Set-BrokerAppAssignmentPolicyRule](#)

[Rename-BrokerAppAssignmentPolicyRule](#)

[Remove-BrokerAppAssignmentPolicyRule](#)

[Add-BrokerUser](#)

# about\_Broker\_Concepts

Jan 04, 2017

## TOPIC

Citrix Broker - Concepts

## SHORT DESCRIPTION

Overview of the Citrix Broker.

## LONG DESCRIPTION

The Citrix Broker is a Microsoft Windows service running on a delivery controller that responds to desktop/application launch requests from users through StoreFront by selecting a suitable machine, powering it up if necessary, and then returning the address of the selected machine to the user's endpoint system so that a session connection can be made. If required for resilience or scale, additional instances of the Broker may be installed to run on additional delivery controllers in the same delivery site.

In addition to handling launch requests, the Broker also has background responsibilities in the delivery site; these include: maintaining appropriate numbers of unused, powered-up machines to avoid unnecessary delays to users launching desktops/applications; maintaining periodic contact with powered-up machines; and monitoring the state of machines and user sessions.

The Citrix.Broker.Admin PowerShell snap-in provides the cmdlets needed to administer and monitor the behaviour of the Broker service, either on the local system (by default) or on another system (by use of the `-AdminAddress` command-line parameter).

The cmdlets in the broker SDK manage objects in the following broad functional areas:

## PROVISIONING CONFIGURATION

The Broker must be informed of the machines which are at its disposal. In order to do this, machines are organized in catalogs, created with the `New-BrokerCatalog` cmdlet; machines are introduced into the system through the use of the `New-BrokerMachine` cmdlet.

Catalogs define the nature of the machines contained within them, such as the allocation type (that is, static or random), how the machines are actually provisioned (that is, by MCS, PVS or manually), whether they are physical or virtual machines, whether they are single-session or multi-session, etc.

Typically, machines configured from a provisioning standpoint are not associated with specific users (though they may need to be, for example if the machine's disk image was captured from a specific user's physical desktop using a P2V tool); this is normally done when configuring how resources are brokered to users, below.

It is also possible for a catalog to be configured to be populated automatically with end users' existing physical machines using the RemotePC feature. The `about_Broker_RemotePC` topic gives more detail.

Provisioning configuration involves the following SDK objects:

`BrokerHypervisorConnection`

BrokerCatalog  
BrokerMachine  
BrokerRemotePCAccount  
BrokerUser

For more information, see:

Get-Help about\_Broker\_Machines  
Get-Help about\_Broker\_RemotePC  
Get-Help Get-BrokerHypervisorConnection  
Get-Help Get-BrokerCatalog  
Get-Help Get-BrokerMachine  
Get-Help Get-BrokerUser

## BROKERING CONFIGURATION

In order that resources (that is, desktops and applications) can be used in user sessions, the Broker must be configured to connect incoming user launch requests through StoreFront with the correct machine. This is achieved by adding machines to desktop groups. The grouping of machines in desktop groups need not necessarily match the grouping of the machines within the catalogs that were used for the configuration of provisioning. It is through the desktop group that the configuration of which users can use which machine resources is achieved.

Configuration of the mapping between resources and end users is achieved through a combination of machine assignment and entitlement rules. In addition, access to those resources must also be configured (for example, some resources could be configured only to be accessible to users when they are not connecting remotely through Access Gateway.) The about\_Broker\_Policies topic gives more detail of the rich configuration options available.

It is also possible for a desktop group to be configured to be populated automatically with end users existing physical machines using the RemotePC feature. The about\_Broker\_RemotePC topic gives more detail.

When machines are virtual, the broker can be configured to minimize power usage by switching them off when they are not expected to be in use while still maintaining good response times for end-user launch requests. This is achieved through power policy for desktop groups. This allows separate configuration for peak compared to off-peak times of the week of the number of machines nominally to be powered up, the number of machines to be powered up and idling, in addition to those in use to be used as a "buffer" to ensure prompt servicing of user launch requests, and the behavior required for virtual machines when users disconnect from their sessions for extended periods of time.

It is also possible to issue explicit power commands to machines. The about\_Broker\_PowerManagement topic gives more detail.

Configuration of Brokering involves the following SDK objects:

BrokerDesktopGroup  
BrokerPrivateDesktop  
BrokerSharedDesktop  
BrokerRemotePCAccount

BrokerPowerTimeScheme  
BrokerUser  
BrokerTag  
BrokerAccessPolicyRule  
BrokerAssignmentPolicyRule  
BrokerEntitlementPolicyRule  
BrokerApplication  
BrokerApplicationInstance  
BrokerAppAssignmentPolicyRule  
BrokerAppEntitlementPolicyRule  
BrokerConfiguredFTA  
BrokerImportedFTA

For more information, see:

Get-Help about\_Broker\_Desktops  
Get-Help about\_Broker\_Policies  
Get-Help about\_Broker\_Applications  
Get-Help about\_Broker\_RemotePC  
Get-Help Get-BrokerPrivateDesktop  
Get-Help Get-BrokerSharedDesktop  
Get-Help Get-BrokerPowerTimeScheme  
Get-Help Get-BrokerUser  
Get-Help Get-BrokerTag  
Get-Help Get-BrokerAccessPolicyRule  
Get-Help Get-BrokerAssignmentPolicyRule  
Get-Help Get-BrokerEntitlementPolicyRule

## MONITORING AND ADMINISTRATION

After you have provisioned and configured machines for brokering, use the broker SDK to monitor and administer user sessions and other aspects of the delivery site.

Monitoring and administration involve the following SDK objects:

BrokerServiceStatus  
BrokerHypervisorAlert  
BrokerDesktop  
BrokerDesktopUsage  
BrokerHostingPowerAction  
BrokerSession  
BrokerSessionMessage

For more information, see:



Get-Help about\_Broker\_Desktops  
Get-Help Get-BrokerServiceStatus  
Get-Help Get-BrokerHypervisorAlert  
Get-Help Get-BrokerDesktop  
Get-Help Get-BrokerDesktopUsage  
Get-Help Get-BrokerHostingPowerAction  
Get-Help Get-BrokerSession  
Get-Help Send-BrokerSessionMessage

## SITE MANAGEMENT

The broker must be configured after installation; this is normally performed automatically by the Citrix Studio console. Configuration tasks include selecting the database (and obtaining the SQL scripting to initialize it), selecting the Citrix Configuration Service that holds the site configuration.

Note that some aspects of broker configuration (such as the port number on which the broker listens for SDK connections) cannot be configured with PowerShell cmdlets. These are configured through the use of the Broker Service executable. For more information, see `about_Broker_PostInstallPreConfiguration`.

A further important aspect of site management concerns the way in which machines providing resources identify the delivery controllers to which they belong. For more information, see `about_Broker_ControllerDiscovery`.

Managing XenDesktop sites involves the following SDK objects:

BrokerSite  
BrokerController  
BrokerDBConnection  
BrokerDBSchema  
BrokerDBVersionChangeScript  
BrokerInstalledDbVersion  
BrokerServiceInstance  
BrokerServiceGroupMembership  
BrokerNameCache

For more information, see:

Get-Help about\_Broker\_PostInstallPreConfiguration  
Get-Help about\_Broker\_ControllerDiscovery  
Get-Help Get-BrokerSite  
Get-Help Get-BrokerController  
Get-Help Get-BrokerDBConnection  
Get-Help Get-BrokerDBSchema  
Get-Help Get-BrokerDBVersionChangeScript  
Get-Help Get-BrokerInstalledDbVersion  
Get-Help Get-BrokerServiceInstance

Get-Help Reset-BrokerServiceGroupMembership  
Get-Help Update-BrokerNameCache

# about\_Broker\_ConfigurationSlots

Jan 04, 2017

## TOPIC

Citrix Broker SDK - Configuration Slots and Machine Configurations

## SHORT DESCRIPTION

Overview of assigning a collection of related settings to a desktop group.

## LONG DESCRIPTION

Collections of related settings may be applied to individual desktop groups through the creation of configuration slots and machine configurations.

A configuration slot defines a collection of related settings that are to be associated with that slot. Each machine configuration is associated with a single configuration slot and provides specific values for settings of that slot.

The SettingsGroup property of the configuration slot determines the particular collection of related settings that are associated with that slot. These groups are defined by Citrix and are not modifiable by administrators. For example, there is a particular group of Profile management specific settings that may be associated with a configuration slot. Because of the close association between a configuration slot and its collection of related settings, the full set of configuration slots is created during the site creation.

Each machine configuration is associated with a single configuration slot. The machine configuration contains policy data that provides specific values for the settings associated with that configuration slot.

Every value set in a machine configuration's policy must belong to the configuration slot's settings group. Therefore the appropriate SDK snap-in must be used to create the policy data. For example, the Profile management snap-in must be used to create the policy data for a machine configuration associated with the Profile management configuration slot.

To have particular policy settings applied to the machines in a desktop group, a machine configuration is associated with that desktop group. A machine configuration may be associated with multiple desktop groups. A desktop group may be associated with multiple machine configurations.

When a machine configuration is associated with a desktop group, the configuration inherits the delegated administration restrictions of the desktop group. Thus, if a machine configuration is associated with multiple desktop groups, an administrator can only modify the policy data of the configuration if the administrator has permission to modify every one of those desktop groups.

For detailed information about defining and assigning machine configurations, see:

`help New-BrokerMachineConfiguration`

`help Add-BrokerMachineConfiguration`

## SEE ALSO

[New-BrokerConfigurationSlot](#)

New-BrokerMachineConfiguration

Add-BrokerMachineConfiguration

Import-BrokerDesktopPolicy

# about\_Broker\_ControllerDiscovery

Jan 04, 2017

## TOPIC

Citrix Broker - Configuring Controller Discovery

## SHORT DESCRIPTION

Describes the way that machines providing published resources discover delivery controllers.

## LONG DESCRIPTION

In order for the broker to be able to connect users to desktops and applications, the machines from which they are published must register (that is, establish communication) with the broker on an appropriate delivery controller in the delivery site.

The default operation, whose configuration is described in this topic, is to use information from the registry. This is referred to as registry-based controller discovery. The registry information can be supplied when installing the delivery agent software on each machine or it can be supplied through group-policy.

If machines are provisioned using quick deploy, information about delivery controllers is stored in a special "identity disk" attached to the VM.

Finally, in some deployments, the use of an Organizational Unit (OU) in Active Directory (AD) may be preferred. This is referred to as AD-based controller discovery. In this case, you must configure the GUID of the OU in the machines' registries. Such configuration is not described in this topic.

To perform registry-based controller discovery, run the PowerShell script called Set-ADControllerDiscovery.ps1 that is installed on each controller in the folder:

```
$Env:ProgramFiles\Citrix\Broker\Service\Setup Scripts
```

For more information, run this script with the -help parameter.

# about\_Broker\_Desktops

Jan 04, 2017

## TOPIC

Citrix Broker SDK - Desktops

## SHORT DESCRIPTION

Describes desktop concepts and usage.

## LONG DESCRIPTION

A desktop is a machine that is able to run a Microsoft Windows desktop environment (with a shell, icons and taskbar) or individual applications (seamlessly integrated with the local desktop). The configuration of the desktop determines whether it can run only desktop environments, only applications, or both desktops and applications. Machines running workstation operating systems are able to run one session at a time (single-session), whereas machines running server operating systems have the ability to run multiple simultaneous sessions (multi-session).

A key aspect of desktops is how they are assigned (or allocated) to users. Two allocation types are supported:

- o Random/Shared - A user is assigned a desktop at random from a pool of shared desktops. Multi-session desktops are able to run sessions to multiple users simultaneously, whereas single-session desktops can only run one session at a time, and are returned to the pool when the user logs off. Single-session shared desktops usually discard user data stored on them after the user logs off. Multi-session shared desktops, however, do not tend to discard user data after a log-off, as this is only possible when the desktop is rebooted by a reboot schedule.
- o Permanent/Private - A private desktop is permanently assigned to a specific user and data stored on it is retained across logons and restarts. A private desktop can have users assigned explicitly or on first use.

## DESKTOP GROUPS

Desktops are collected together in desktop groups, and these provide a flexible grouping mechanism that can be used to associate:

- o Desktops running on a particular type of machine
- o Desktops with particular software installed
- o Desktops for a set of users
- o Desktops accessed in a similar way
- o Desktops configured in a particular way
- o Any combination of the above

Each desktop group can only contain one type of desktop, determined by its `AllocationType` and `SessionSupport` properties.

When assigning shared desktops or assign-on-first-use (AOFU) desktops to users, the set of candidates comes from available desktops in a particular desktop group.

You configure power management policy for single-session desktop groups, including peak and off-peak settings, for each desktop group. See `about_Broker_PowerManagement` for details.

## CREATION OF DESKTOPS

Desktop objects are created automatically when a machine is added to a desktop group. The type of desktop is determined by the `AllocationType` property of the desktop group.

In order for a machine to be added from a catalog, the machine must be compatible with the desktop group. For this to be true, the catalog's `AllocationType` must be compatible, and the `SessionSupport` property must match.

Note: Because the session support and functional level of the machine are determined by the software on the machine (operating system and Citrix VDA, respectively), the `SessionSupport` and `MinimalFunctionalLevel` of the catalog and desktop group may match, but not be compatible with the machine. In this case any attempt of registration by the machine will fail.

You can add machines explicitly using the `Add-BrokerMachine` cmdlet, or a number of free machines can be acquired from a catalog using the `Add-BrokerMachinesToDesktopGroup` cmdlet. A machine can only be associated with one desktop group at a time, and has a `DesktopUid` property that references the corresponding desktop object. You can also associate desktops to machines with their SID properties.

Desktop objects are deleted when the machine is removed from the desktop group (using the `Remove-BrokerMachine` cmdlet). Desktops are also deleted when the desktop group containing them is deleted (using the `Remove-BrokerDesktopGroup` cmdlet). A desktop cannot be removed from a catalog while it is in a desktop group.

## SHARED (RANDOM) DESKTOPS

Shared desktops are published to users using entitlement policy rules. Each entitlement rule allows access to a single session on a desktop machine, selected at random from the available desktops in a desktop group (with a preference for desktops that are powered on). If there are no available desktops, launching the session fails. See `about_Broker_EntitlementPolicy` and `about_Broker_PowerManagement` for details.

## PRIVATE (STATIC) DESKTOPS

You can assign private desktops to users explicitly or automatically, with the AOFU feature. It is also possible to assign private desktops to particular clients (through IP or client name).

You explicitly assign machines or desktops to users with the `Add-BrokerUser` cmdlet. Machines can be assigned to users before the machine has been added to the desktop group (desktop created), but otherwise the effect is the same.

With `Add-BrokerUser` you can assign a desktop to multiple users or user groups. If the desktop has single-session support then the desktop will be visible to multiple users, but only one user can log on to the desktop at any time.

Assignment policy rules allow you to use AOFU to assign desktops to users in a desktop group. When a user specified in an assignment policy rule launches a session, and if the user does not already have an assigned desktop, the broker selects an available desktop at random from the desktop group and permanently assigns it to that user. Once assigned in this way, the

desktop behaves as though the assignment was made explicitly by an administrator. See about\_Broker\_AssignmentPolicy for details.

It is possible to assign more than one desktop to a user. You can achieve this by explicit assignment, multiple assignment policy rules, or the MaxDesktops property of an assignment policy rule.

## DESKTOP CONFIGURATION

When presented to the user, desktops are identified by:

- o An icon (IconUid property)
- o A name (PublishedName property)
- o A description (Description property)

When starting the session, you can configure two connection settings:

- o The color depth used at the start of the session (ColorDepth property)
- o Whether SecureICA encryption is required (SecureIcaRequired property)

Each desktop group provides default values for these settings, but you can override them if the desktop has more specific settings (PrivateDesktop), or use an entitlement policy rule with more specific settings (SharedDesktop).

AOFU desktops can inherit these settings from the assignment policy rule when the assignment to the user takes place.

## MAINTENANCE MODE

There are times when it is necessary to disable desktops. You can do this by setting the InMaintenanceMode property of a desktop to \$true. This puts it into maintenance mode. The broker excludes single-session desktops in maintenance mode from brokering decisions and does not start new sessions on them. Existing sessions are unaffected. For multi-session desktops in maintenance mode, reconnections to existing sessions are allowed, but no new sessions are created on the machine.

Desktops in maintenance mode are also excluded from automatic power management, although explicit power actions are still performed.

Note that disabling desktop groups, entitlement policy rules, assignment policy rules, or applications are other ways of disabling aspects of brokering.

## DESKTOP STATUS

Once desktops are created, you can query the configuration and state information for different kinds of desktop, or retrieve more information about desktops using the Get-BrokerMachine cmdlet. To get details of any sessions running on the desktops, use the Get-BrokerSession cmdlet.

You can also group desktops by a specific property, counting the number of desktops with each value using the Group-BrokerMachine cmdlet. This can provide useful summary statistics.

## DESKTOP USAGE



Every hour the broker records how many desktops from each desktop group are in use, and the `Get-BrokerDesktopUsage` cmdlet returns this information. Analyze historical usage records to understand desktop usage patterns and help with the choice of idle pool and buffer settings.

## DESKTOP CONDITIONS

CPU usage, ICA latency, and profile logon times of desktops are monitored. When one of these values exceeds a threshold (configured by policy), the condition is flagged in the `DesktopConditions` property of the desktop. When the value drops below the threshold again, the condition is cleared. Use `Get-BrokerMachine` or `Group-BrokerMachines` cmdlets to query this information.

## SEE ALSO

[about\\_Broker\\_Concepts](#)

[about\\_Broker\\_Applications](#)

[about\\_Broker\\_EntitlementPolicy](#)

[about\\_Broker\\_AssignmentPolicy](#)

[Add-BrokerMachine](#)

[Add-BrokerMachinesToDesktopGroup](#)

[Remove-BrokerMachine](#)

[Add-BrokerUser](#)

[Set-BrokerMachine](#)

[Set-BrokerPrivateDesktop](#)

[Get-BrokerMachine](#)

[Group-BrokerMachine](#)

[Get-BrokerDesktopUsage](#)

# about\_Broker\_EntitlementPolicy

Jan 04, 2017

## TOPIC

Citrix Broker SDK - Desktop and Application Entitlement Policy

## SHORT DESCRIPTION

Controls end-user entitlement to desktop and application sessions provided from a pool of shared machines.

## LONG DESCRIPTION

The site's entitlement policy defines rules controlling users' entitlements to desktop and application sessions from pools of shared machines. Each pool is defined by a desktop group.

The entitlement policy comprises a set of rules. Each rule grants users a single entitlement to a desktop or application session in a specified desktop group. The policy can be applied only to groups of desktop kind Random. For desktop entitlements multiple rules can apply to the same group, however for application entitlements only a single rule can apply to a given group.

When the user starts a session by selecting an entitlement the behavior depends on the session-support property of the desktop group:

- o For single-session groups the user is temporarily assigned a machine selected at random from the group to provide their session. When the session ends, the machine is returned to the pool of available machines.
- o For multi-session groups the user session is provided by the machine that is least loaded within the group when the session is launched.

If multiple desktop entitlement rules for the same group contain the same user, the user can have as many desktop sessions from the group concurrently as they have entitlements.

Although only a single application entitlement rule can be defined for a group, a user can still launch multiple applications from that group because the applications all run within that entitlement's single session.

Rules for desktop and application session entitlements are distinct. Desktop entitlements are managed through the BrokerEntitlementPolicyRule SDK object, and application rules through the BrokerAppEntitlementPolicyRule object.

Desktop entitlement rules can be created only for desktop groups with delivery types DesktopsOnly or DesktopsAndApps, whereas an application entitlement rule can be created only for delivery types AppsOnly or DesktopsAndApps.

For desktop groups with delivery type DesktopsAndApps, typically one or more desktop session entitlement rules together with a single application session entitlement rule exist.

For an entitlement granted by the entitlement policy to be available to a user, the site's access policy must also grant them access to the desktop group.

## ENTITLEMENT POLICY RULES

Each entitlement policy rule has the following key properties:

- o The desktop group to which it applies (one rule only ever applies to one group)
- o The users to whom the entitlement is granted

Additionally for desktop entitlement rules, the following properties exist:

- o The published name of the entitlement (visible to the user)
- o Any properties that a desktop session launched using the entitlement should use that differ from the defaults specified on the desktop group

If multiple desktop entitlements are available to a user from the same group the resultant desktop session properties may differ depending on which entitlement the user selects to start the session.

Each rule can be individually enabled or disabled. A disabled rule is ignored when the entitlement policy is evaluated.

#### USER FILTERS (FULL)

Each rule has two user filters, an include filter and an exclude filter:

- o The include filter contains users and user groups that are granted an entitlement to a session
- o The exclude filter contains users and user groups that are denied an entitlement to a session

If the include filter of a rule contains multiple instances of a user (either explicit or implicit), they get only one entitlement by that rule.

Entries in the exclude filter take priority, so if a user appears explicitly or implicitly in both filters, access is denied. Typically, you use this filter to exclude a user or group of users who would otherwise gain access because they are members of a user group specified in the include filter.

Because all rules are independently evaluated, the exclude filter can only exclude users who would otherwise gain an entitlement through the same rule's include filter. That is, if a user is in a rule's include filter but not its exclude filter, the rule is guaranteed to grant that user a session entitlement irrespective of whether the user appears in the exclude filter of other rules.

If a filter contains a user group that contains other users and groups, the filter implicitly includes all of those users and groups.

By default the exclude filter is disabled.

To maintain entitlement policy rules that can be fully displayed and edited with Citrix Studio, use the simplified user filter model below and do not use the exclude filter.

## USER FILTERS (SIMPLIFIED)

The included user filter described above also supports a simplified usage model where the filter itself is disabled. When this is done, any user who has access to the desktop group through the access policy is implicitly granted an entitlement to a session through the entitlement policy rule without the need to list the user in the rule's include filter.

This is useful in cases where the access policy for the desktop group already explicitly specifies the users who should have access.

Even if the include filter is disabled, the exclude filter can still be used to deny the entitlement from users who would otherwise gain access through the access policy alone.

## ADDITIONAL DESKTOP ENTITLEMENT RULE PROPERTIES

Desktop entitlement rules specify the following additional properties:

- o PublishedName
- o Description
- o IconUid
- o ColorDepth
- o SecureIcaRequired

The published name, description, and icon UID properties apply to the desktop entitlement itself and determine the way in which the entitlement is presented to the user in, for example, StoreFront.

The color depth and secure ICA properties apply to the desktop session that is obtained when the entitlement is used.

In all cases, these properties can be explicitly specified. However, a null value (the default) means that the corresponding property is taken from the desktop group to which the rule applies. This inheritance from groups is dynamic; if the property of the group changes, the property of the entitlement changes too.

## NOTES

If a rule grants an entitlement to a user group, the session entitlement applies to the individual user who selects the entitlement. However, this does not prevent a different user in the same user group from using the same entitlement concurrently. So, a rule that grants an entitlement to a user group containing multiple users allows each user concurrent access to a single session from the desktop group.

The total number of entitlements defined by the policy may exceed the number of machines available, or the maximum allowed sessions, from the desktop group. A user attempting to use an entitlement when no further resources are available receives a no-desktop-available error.

If a session launched through an entitlement is active when the entitlement rule is deleted, the session continues unaffected. However:

- o When the user ends the session, they cannot start a new one if the deleted rule was their only entitlement to a session in that group
- o If the user disconnects the session, they cannot reconnect to it

## SEE ALSO

[about\\_Broker\\_Policies](#)

[about\\_Broker\\_AccessPolicy](#)

[about\\_Broker\\_AssignmentPolicy](#)

[about\\_Broker\\_Applications](#)

[New-BrokerEntitlementPolicyRule](#)

[Get-BrokerEntitlementPolicyRule](#)

[Set-BrokerEntitlementPolicyRule](#)

[Rename-BrokerEntitlementPolicyRule](#)

[Remove-BrokerEntitlementPolicyRule](#)

[New-BrokerAppEntitlementPolicyRule](#)

[Get-BrokerAppEntitlementPolicyRule](#)

[Set-BrokerAppEntitlementPolicyRule](#)

[Rename-BrokerAppEntitlementPolicyRule](#)

[Remove-BrokerAppEntitlementPolicyRule](#)

# about\_Broker\_ErrorHandling

Jan 04, 2017

## TOPIC

Citrix Broker SDK - Error Handling

## SHORT DESCRIPTION

Describes broker errors generated by cmdlets and how to access them.

## LONG DESCRIPTION

The broker SDK cmdlets report errors through the class `SdkErrorRecord`, which is a subclass of the standard Powershell error record class `System.Management.Automation.ErrorRecord`. `SdkErrorRecord` contains:

- o A short string to describe the error status code. This is implemented as a public property named `Status`.
- o A dictionary of key-value pairs containing additional data specific to the cmdlet. This is implemented as a public property named `ErrorData` of type `Dictionary<string, string>`.

The error status property always has a value. Populating the error data dictionary is optional. The number of entries within the dictionary, the content of the entries, and the exact format of the key and value data is specific to each cmdlet.

You can access an `SdkErrorRecord` object using the standard Powershell cmdlet `ErrorVariable` parameter. The type of object returned by `ErrorVariable` depends on whether the error is terminating or non-terminating.

## NON-TERMINATING ERRORS

For non-terminating errors, each object in the returned `ErrorVariable` array is simply an instance of type `SdkErrorRecord`.

## TERMINATING ERRORS

For terminating errors, the object returned by `ErrorVariable` is of type `System.Management.Automation.CmdletInvocationException`.

For non-terminating errors that are escalated as terminating errors (through the "ErrorAction stop" argument), the object returned by `ErrorVariable` is of type `System.Management.Automation.ActionPreferenceStopException`.

`CmdletInvocationException` and `ActionPreferenceStopException` are subclasses of the base class `System.Management.Automation.RuntimeException`, which exposes the `SdkErrorRecord` object through its `ErrorData` property.

## Class `SdkOperationException`

`SdkErrorRecord`'s `Exception` property holds an instance of custom exception class `SdkOperationException`, which also contains the error status code and data dictionary from `SdkErrorRecord`.

The SDK cmdlets generate errors in response to exceptions generated by the underlying system or by the cmdlet detecting

errors locally and instantiating appropriate exception types. Such exceptions, which represent the original cause of the terminating error, are specified in `SdkOperationException`'s `InnerException` property.

For terminating errors, use Powershell scripts to trap `SdkOperationException` and access additional error information and the originating exception.

## REVIEW OF POWERSHELL ERROR HANDLING BEHAVIOR

Powershell scripts can access error information using the following methods:

- o Read error records from the `$Error` arraylist.
- o Get the cmdlet to return error records using the standard `ErrorVariable` cmdlet parameter.
- o Use trap blocks to catch exceptions for terminating errors.

The type of the error record that Powershell puts in the `$Error` arraylist and returns through the `ErrorVariable` cmdlet parameter depends on whether the error is terminating or non-terminating, and whether the "ErrorAction continue" or "ErrorAction stop" cmdlet parameters are specified (that turn terminating errors into non-terminating ones, and vice versa).

For the combinations of error types (terminating, non-terminating) and error actions (stop, continue), the following statements describe the relationship between:

- o The type of object contained in the Powershell `$Error` arraylist.
- o The type of object returned by the `ErrorVariable` cmdlet parameter (assuming the script can continue after a terminating error).
- o The type of the exception that is thrown (where applicable).

Non-terminating errors with `ErrorAction=Continue`, `$Error` type=`SdkErrorRecord`, and `ErrorVariable` type=`SdkErrorRecord` do not throw an exception.

Terminating errors with `ErrorAction=Stop`, `$Error` type=`ErrorRecord`, and `ErrorVariable` type=`CmdletInvocationException` throw an `SdkOperationException` exception.

Non-terminating errors with `ErrorAction=Stop`, `$Error` type=`ErrorRecord`, and `ErrorVariable` type=`ActionPreferenceOperationException` do not throw an exception.

Terminating errors with `ErrorAction=Continue`, `$Error` type=`ErrorRecord`, and `ErrorVariable` type=`CmdletInvocationException` throw an `SdkOperationException` exception.

`ActionPreferenceOperationException` and `CmdletInvocationException` are subclasses of `System.Management.Automation.RuntimeException`.

## ACCESSING ERROR RECORD DATA

The Powershell code sample below demonstrates how to access error information programmatically with the `ErrorVariable` cmdlet parameter and a trap block.

```
# Trap exceptions generated from terminating errors trap [Exception] {
```

```

write ""
write "TRAP BLOCK : BEGIN"

if($_.Exception.GetType().Name -eq "SdkOperationException")
{
    $sdkOpEx = $_.Exception

    # show error status
    write $("SdkOperationException.Status = " + $sdkOpEx.Status)

    # show error data dictionary
    write $("SdkOperationException.ErrorData=")

    write $("SdkOperationException.InnerException = " + $sdkOpEx.InnerException)
    $_.Exception.ErrorData
}

continue #could also call break here to halt script execution
write "TRAP BLOCK : END"

}

##### ## Run tests 1 to
4, below, in turn to examine terminating and ## non-terminating error behavior.
#####

## Test 1: Invoke cmdlet that generates a terminating error: # New-BrokerCatalog throws terminating error if a # catalog
with the supplied name already exists. # #New-BrokerCatalog -Name "AlreadyExists" -AllocationType Random -
ProvisioningType Manual -SessionSupport SingleSession -PersistUserChanges OnLocal -MachinesArePhysical $true -
ErrorVariable ev

## Test 2: Force script execution to continue after a terminating error. # #New-BrokerCatalog -Name "AlreadyExists" -
AllocationType Random -ProvisioningType Manual -SessionSupport SingleSession -PersistUserChanges OnLocal -
MachinesArePhysical $true -ErrorVariable ev -ErrorAction continue

## Test 3: Invoke cmdlet that generates a non-terminating error: # Get-BrokerCatalog generates a non-terminating error if
a catalog # with the specified name doesn't exist. # #Get-BrokerCatalog -Name "IDontExist" -ErrorVariable ev

## Test 4 Force script execution to halt after a non-terminating error. # #Get-BrokerCatalog -Name "IDontExist" -
ErrorVariable ev -ErrorAction "stop"

write "" write "GET ERROR INFORAMTION: BEGIN"

$SdkErrRecord = $null

if($ev[0].GetType().Name -eq "SdkErrorRecord"){

```



```

    $sdkErrRecord = $ev[0]
}
elseif($ev[0].GetType().BaseType.FullName -eq "System.Management.Automation.RuntimeException") {

    $sdkErrRecord = $ev[0].ErrorRecord

} else {

    write ("UNKNOWN ERROR VARIABLE TYPE:")
    $ev[0].GetType().Name

}

if($sdkErrRecord -ne $null) {

    write ("Have sdkErrRecord:")
    write (" Type Name = " + $sdkErrRecord.GetType().FullName)
    write (" Status = " + $sdkErrRecord.Status)
    write (" Exception type = " + $sdkErrRecord.Exception.GetType().FullName)
    write (" ErrorData = ")
    $sdkErrRecord.ErrorData
    write (" FullyQualifiedErrorId = " + $sdkErrRecord.FullyQualifiedErrorId)

} write "GET ERROR INFORMATION: END"

```

# about\_Broker\_Filtering

Jan 04, 2017

## TOPIC

XenDesktop - Advanced Dataset Filtering

## SHORT DESCRIPTION

Describes the common filtering options for XenDesktop cmdlets.

## LONG DESCRIPTION

Some cmdlets operate on large quantities of data and, to reduce the overhead of sending all of that data over the network, many of the Get- cmdlets support server-side filtering of the results.

The conventional way of filtering results in PowerShell is to pipeline them into Where-Object, Select-Object, and Sort-Object, for example:

```
Get-<Noun> | Where { $_.Size = 'Small' } | Sort 'Date' | Select -First 10
```

However, for most XenDesktop cmdlets the data is stored remotely and it would be slow and inefficient to retrieve large amounts of data over the network and then discard most of it. Instead, many of the Get- cmdlets provide filtering parameters that allow results to be processed on the server, returning only the required results.

You can filter results by most object properties using parameters derived from the property name. You can also sort results or limit them to a specified number of records:

```
Get-<Noun> -Size 'Small' -SortBy 'Date' -MaxRecordCount 10
```

You can express more complex filter conditions using a syntax and set of operators very similar to those used by PowerShell expressions.

Those cmdlets that support filtering have the following common parameters:

`-MaxRecordCount <int>`

Specifies the maximum number of results to return.  
For example, to return only the first nine results use:

```
Get-<Noun> -MaxRecordCount 9
```

If not specified, only the first 250 records are returned, and if more are available, a warning is produced:

WARNING: Only first 250 records returned. Use -MaxRecordCount to retrieve more.

You can suppress this warning by using -WarningAction or by specifying a value for -MaxRecordCount.

To retrieve all records, specify a large number for -MaxRecordCount. As the value is an integer, you can use the following:

```
Get-<Noun> -MaxRecordCount [int]::MaxValue
```

-ReturnTotalRecordCount [<SwitchParameter>]

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. For example:

```
Get-<Noun> -MaxRecordCount 9 -ReturnTotalRecordCount
```

....

```
Get-<Noun> : Returned 9 of 10 items
```

```
At line:1 char:18
```

```
+ Get-<Noun> <<<< -MaxRecordCount 9 -ReturnTotalRecordCount
```

```
+ CategoryInfo          : OperationStopped: (:) [Get-<Noun>], PartialDataException
```

```
+ FullyQualifiedErrorId : PartialData,Citrix.<SDKName>.SDK.Get<Noun>
```

The count can be accessed using the TotalAvailableResultCount property:

```
$count = $error[0].TotalAvailableResultCount
```

-Skip <int>

Skips the specified number of records before returning results.

Also reduces the count returned by -ReturnTotalRecordCount.

-SortBy <string>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order, respectively. Ascending order is assumed if no prefix is present.

Sorting occurs before -MaxRecordCount and -Skip parameters are applied. For example, to sort by Name and then by Count (largest first) use:

```
-SortBy 'Name,-Count'
```

By default, sorting by an enumeration property uses the numeric value of the elements. You can specify a different sort order by qualifying the name with an ordered list of elements or their numeric values, or <null> to indicate the placement of null values.

Elements not mentioned are placed at the end in their numeric order. For example, to sort by two different enums and then by the object id:

```
-SortBy 'MyState(StateC,<null>,StateA,StateB),Another(0,3,2,1),Id'
```

-Filter <String>

This parameter lets you specify advanced filter expressions, and supports combination of conditions with -and and -or, and grouping with braces. For example:

```
Get-<Noun> -Filter 'Name -like "High*" -or (Priority -eq 1 -and Severity -ge 2)'
```

The syntax is close enough to PowerShell syntax that you can use script blocks in most cases. This can be easier to read as it reduces quoting:

```
Get-<Noun> -Filter { Count -ne $null }
```

The full -Filter syntax is provided below.

## EXAMPLES

Filtering by strings performs a case-insensitive wildcard match.

Separate parameters are combined with an implicit -and operator.

Normal PowerShell quoting rules apply, so you can use single or double quotes, and omit the quotes altogether for many strings. The order of parameters does not make any difference. The following are equivalent:

```
Get-<Noun> -Company Citrix -Product Xen*
Get-<Noun> -Company "citrix" -Product '[X]EN*'
Get-<Noun> -Product "Xen*" -Company "CITRIX"
Get-<Noun> -Filter { Company -eq 'Citrix' -and Product -like 'Xen*' }
```

See [about\\_Quoting\\_Rules](#) and [about\\_Wildcards](#) for details about PowerShell handling of quotes and wildcards.

To avoid wildcard matching or include quote characters, you can escape the wildcards using the normal PowerShell escape mechanisms (see [about\\_Escape\\_Characters](#)), or switch to a filter expression and the `-eq` operator:

```
Get-<Noun> -Company "Abc[*]"           # Matches Abc*
Get-<Noun> -Company "Abc`*"           # Matches Abc*
Get-<Noun> -Filter { Company -eq "Abc*" } # Matches Abc*
Get-<Noun> -Filter { Company -eq "A`B`C" } # Matches A"B'C
```

Simple filtering by numbers, booleans, and TimeSpans perform direct equality comparisons, although if the value is nullable you can also search for null values. Here are some examples:

```
Get-<Noun> -Uid 123
Get-<Noun> -Enabled $true
Get-<Noun> -Duration 1:30:40
Get-<Noun> -NullableProperty $null
```

More comparisons are possible using advanced filtering with `-Filter`:

```
Get-<Noun> -Filter 'Capacity -ge 10gb'
Get-<Noun> -Filter 'Age -ge 20 -and Age -lt 40'
Get-<Noun> -Filter 'VolumeLevel -like "[123]"'
Get-<Noun> -Filter 'Enabled -ne $false'
Get-<Noun> -Filter 'NullableProperty -ne $null'
```

You can check boolean values without an explicit comparison operator, and you can also combine them with `-not`:

```
Get-<Noun> -Filter 'Enabled' # Equivalent to 'Enabled -eq $true'
Get-<Noun> -Filter '-not Enabled' # Equivalent to 'Enabled -eq $false'
```

See [about\\_Comparison\\_Operators](#) for an explanation of the operators, but note that only a subset of PowerShell operators are supported (`-eq`, `-ne`, `-gt`, `-ge`, `-lt`, `-le`, `-like`, `-notlike`, `-in`, `-notin`, `-contains`, `-notcontains`).

Enumeration values can either be specified using typed values or the string name of the enumeration value:

```
Get-<Noun> -Shape [Shapes]::Square
Get-<Noun> -Shape Circle
```

With filter expressions, typed values can be specified with simple variables or quoted strings. They also support enumerations with wildcards:

```
$s = [Shapes]::Square
Get-<Noun> -Filter { Shape -eq $s -or Shape -eq "Circle" }
Get-<Noun> -Filter { Shape -like 'C*' }
```

By their nature, floating point values, DateTime values, and TimeSpan values are best suited to relative comparisons rather than just equality. DateTime strings are converted using the locale and time zone of the user device, but you can use ISO8601 format strings (YYYY-MM-DDThh:mm:ss.sTZD) to avoid ambiguity. You can also use standard PowerShell syntax to create these values:

```
Get-<Noun> -Filter { StartTime -ge "2010-08-23T12:30:00.0Z" }
$d = [DateTime]"2010-08-23T12:30:00.0Z"
Get-<Noun> -Filter { StartTime -ge $d }
$d = (Get-Date).AddDays(-1)
Get-<Noun> -Filter { StartTime -ge $d }
```

Relative times are quite common and, when using filter expressions, you can also specify DateTime values using a relative format:

```
Get-<Noun> -Filter { StartTime -ge '-2' } # Two days ago
Get-<Noun> -Filter { StartTime -ge '-1:30' } # Hour and a half ago
Get-<Noun> -Filter { StartTime -ge '-0:0:30' } # 30 seconds ago
```

## ARRAY PROPERTIES

When filtering against list or array properties, simple parameters perform a case-insensitive wildcard match against each of the members. With filter expressions, you can use the -contains and -notcontains operators. Unlike PowerShell, these perform wildcard matching on strings.

Note that for array properties the naming convention is for the returned property to be plural, but the parameter used to search for any match is singular. The following are equivalent (assuming Users is an array property):

```
Get-<Noun> -User Fred*
Get-<Noun> -Filter { User -like "Fred*" }
Get-<Noun> -Filter { Users -contains "Fred*" }
```

You can also use the singular form with -Filter to search using other operators:

```
# Match if any user in the list is called "Frederick"
Get-<Noun> -Filter { User -eq "Frederick" }
# Match if any user in the list has a name alphabetically below 'F'
Get-<Noun> -Filter { User -lt 'F' }
```

## COMPLEX EXPRESSIONS

When matching against multiple values, you can use a sequence of comparisons joined with -or operators, or you can use -in and -notin:

```
Get-<Noun> -Filter { Shape -eq 'Circle' -or Shape -eq 'Square' }
$shapes = 'Circle','Square'
Get-<Noun> -Filter { Shape -in $shapes }
$sides = 1..4
Get-<Noun> -Filter { Sides -notin $sides }
```

Braces can be used to group complex expressions, and override the default left-to-right evaluation of -and and -or. You can also use -not to invert the sense of any sub-expression:

```
Get-<Noun> -Filter { Size -gt 4 -or (Color -eq 'Blue' -and Shape -eq 'Circle') }
Get-<Noun> -Filter { Sides -lt 5 -and -not (Color -eq 'Blue' -and Shape -eq 'Circle') }
```

## PAGING

The simplest way to page through data is to use the -Skip and -MaxRecordCount parameters. So, to read the first three pages of data with 10 records per page, use:

```
Get-<Noun> -Skip 0 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 10 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 20 -MaxRecordCount 10 <other filtering criteria>
```

You must include the same filtering criteria on each call, and ensure that the data is sorted consistently.

The above approach is often acceptable, but as each call performs an independent query, data changes can result in records being skipped or appearing twice. One approach to improve this is to sort by a unique id field and then start the search for the next page at the unique id after the last unique id of the previous page. For example:

```
# Get the first page
Get-<Noun> -MaxRecordCount 10 -SortBy SerialNumber

SerialNumber ...
----- ---
A120004
A120007
... 7 other records ...
A120900

# Get the next page
Get-<Noun> -MaxRecordCount 10 -Filter { FirstName -gt 'A120900' }

SerialNumber ...
```

----- ---  
A120901  
B220000  
...

## FILTER SYNTAX DEFINITION

<Filter> ::= <ScriptBlock> | <ComponentList>

<ScriptBlock> ::= "{" <ComponentList> "}"

<ComponentList> ::= <Component> <AndOrOperator> <ComponentList> |

<Component>

<Component> ::= <NotOperator> <Factor> |

<Factor>

<Factor> ::= "(" <ComponentList> ")" |

<PropertyName> <ComparisonOperator> <Value> |  
<PropertyName>

<AndOrOperator> ::= "-and" | "-or"

<NotOperator> ::= "-not" | "!"

<ComparisonOperator>

::= "-eq" | "-ne" | "-le" | "-ge" | "-lt" | "-gt" |  
"-like" | "-notlike" | "-contains" | "-notcontains" |  
"-in" | "-notin"

<PropertyName> ::= <simple name of property>

<Value> ::= <string literal> | <numeric literal> |

<scalar variable> | <array variable> |  
"\$null" | "\$true" | "\$false"



Numeric literals support decimal and hexadecimal literals, with optional multiplier suffixes (kb, mb, gb, tb, pb).

Dates and times can be specified as string literals. The current culture determines what formats are accepted. To avoid any ambiguity, use strings formatted to the ISO8601 standard. If not specified, the current time zone is used.

Relative date-time string literals are also supported, using a minus sign followed by a TimeSpan. For example, "-1:30" means 1 hour and 30 minutes ago.

# about\_Broker\_Licensing

Jan 04, 2017

## TOPIC

Citrix Broker - Licensing

## SHORT DESCRIPTION

Overview of broker licensing configuration.

## LONG DESCRIPTION

As part of the licensing setup for a site, the types of product licenses used by the broker when creating connections to virtual desktops or applications can be configured using the Central Configuration Service SDK.

The following licensing related properties can be specified using the Set-ConfigSite cmdlet:

- o LicensingModel - Sets the licensing model to use. Values can be 'Concurrent' or 'UserDevice'.
- o ProductCode - Specifies which product license is supported by the site. Values can be **◆MPS◆** for XenApp licenses or **◆XDT◆** for XenDesktop licenses.
- o ProductEdition - Sets the licensing edition to use.

A license matching the specified model, product code, and edition must be available within the site's license server in order for the broker to grant licenses.

These properties are part of the site object returned by the Get-ConfigSite cmdlet.

## CONCURRENT LICENSE MODEL

A concurrent license is tied to a XenDesktop session. When a user launches a session, a license is checked out to that session. When a user logs off from a session, the license is checked back in again, making it available for another session.

## USER DEVICE LICENSE MODEL

With user device licensing, the license server automatically assigns licenses to users or devices based on usage:

- o User licensing allows users to access their desktops and applications from multiple devices.
- o Device licensing allows multiple users to access their desktops and applications from a single device.

When users or devices connect to an application or desktop, they consume a license for a 90-day license assignment period. The assignment period begins when a connection is made, is renewed to the full 90 days during the life of the connection, and expires (allowing reassignment) 90 days after the last connection terminates. A license assignment can be manually

ended before the 90-day period elapses using the `udadmin` command line installed on the license server.

## LICENSING STATE

The broker site contains the following properties related to licensing state:

- o `LicensingGracePeriodActive` - Reports if the broker is in licensing grace period.
- o `LicensingOutOfBoxGracePeriodActive` - Reports if the broker is in out-of-box grace period.
- o `LicensingGraceHoursLeft` - The number of grace hours remaining, if the broker is in grace period, else it is null.
- o `LicensedSessionsActive` - The number of active, licensed sessions.
- o `LicenseGraceSessionsRemaining` - The number of grace sessions available, if the broker is in licensing grace period, else it is null.

These properties are part of the site object returned by the `Get-BrokerSite` cmdlet.

## LICENSE SERVER TEST

The broker SDK cmdlet `Test-BrokerLicenseServer` checks whether or not a given server can be used as a license server by the broker.

## RESETTING LICENSE SERVER CONNECTION

The broker SDK cmdlet `Reset-BrokerLicensingConnection` resets the broker's connection to the license server.

## LICENSE BURN-IN DATE

The version of the product that is supported within the site is denoted by a licensing burn-in date. This date can be accessed through the `LicensingBurnInDate` field of the site object returned by the `Get-ConfigSite` cmdlet.

## SEE ALSO

`about_Broker_Site`

[Test-BrokerLicenseServer](#)

[Reset-BrokerLicensingConnection](#)

[Get-BrokerController](#)

`Set-ConfigSite`

`Get-ConfigSite`

[Get-BrokerSite](#)

# about\_Broker\_Machines

Jan 04, 2017

## TOPIC

Citrix Broker SDK - Machine Object

## SHORT DESCRIPTION

Describes machine concepts and usage.

## LONG DESCRIPTION

A machine represents a physical or virtual machine that can be used to provide a user with one or more desktops, applications or both. When a machine is created, you must assign it to a catalog, which defines how the machine is allocated to a user (static or random), the session support it provides (single-session or multi-session), how the machine's disk images are created and managed (PVS, MCS or manually) and the expected functional level. If the machine is virtual but not provisioned by MCS, you must also assign it to a hypervisor connection, which represents the hypervisor (or pool of linked hypervisors) that runs the virtual machine.

Creating a machine object is the first step in the broker SDK towards configuring a physical or virtual machine to provide desktops and/or applications to users. A machine must be added to a desktop group before it is able to be used (see about\_Broker\_Desktops). To add machines to a desktop group, the Add-BrokerMachine or Add-BrokerMachinesToDesktopGroup cmdlets can be used. This creates a desktop object corresponding to the machine.

## CATALOGS

When a machine is created, you must assign it to a catalog. The catalog defines the behavior of the machine within a site as well as the expected functionality and properties of the machine:

- o Allocation type: The catalog determines how the machines are allocated to the user. Allocation can be static or random. Static allocation is where the machine is permanently assigned to a specific user. Data stored is retained across logons and restarts.

The other type of allocation is random, where a random machine is assigned to a user from a pool when a session is requested. The machine returns to the pool when the user logs off.

- o How the machine is created: The catalog collects together machines that are created in the same way: either with PVS, MCS or manually.
- o Physical or Virtual: A machine that is virtual can have its power state controlled and monitored by the system. Virtual machines must be associated with a hypervisor connection, either directly or, in the case of MCS provisioned machines, indirectly through the provisioning scheme. Single session virtual machines can be managed using power policy to automatically be turned on or off as needed. Machines marked as physical are not monitored or controlled as to their power state.

- o How the users settings are stored: The catalog also determines how the users settings are stored, either on a Citrix Personal vDisk, on the machine's local drive, or if the user settings are discarded.
- o RemotePC: If the catalog is set up as a remote PC catalog, machines are added automatically upon registration based on the site configuration. In order for a catalog to be specified as a RemotePC catalog, the session support must be single session and the catalog must be set up for physical machines.
- o Functional level: The functional level of a machine is determined by the version of the Citrix VDA software it is running. Some features are not supported in machines with lower functional levels. Catalogs can supply a minimal functional level, meaning any machines in the catalog with a lower functional level will be unable to register with the site.
- o Session support: This can either be single-session or multi-session. Single-session machines can have an active session with up to one user at any time, whereas multi-session machines have the capability to have active sessions with multiple users simultaneously. The session support of a machine is determined by the variant of the VDA software component installed on the machine (either with single-session support or multi-session support). The multi-session VDA software may only be installed on server operating systems. The catalog session support must match the session support of the software installed on the machine for the machine to successfully register with the site.

## MACHINE STATUS

After machines are created, you can query the configuration and state information using the Get-BrokerMachine cmdlet. The information the cmdlet can provide includes, but is not limited to, the following:

- o Personal vDisk interactions and lifecycle: The current state of the personal vDisk can be obtained, as well as the configuration options of how the user data is persisted.
- o Session properties: The properties of the current session for single-session machines can be obtained, such as ClientName and ClientAddress. To access session information on multi-session machines, the Get-BrokerSession cmdlet can be used.
- o Application status: If the machine is configured to run applications, information can be found about the published applications running on the machine.
- o Connection information: Information about the time and user of the last

connection to the machine can be found, as well as information about the last deregistration.

For an exhaustive list of the properties of a machine that can be queried, see the `Get-BrokerMachine` cmdlet.

## MACHINE CONFIGURATION

Machine settings can be changed and configured once the machine object has been created, as long as the changes are compatible with the catalog the machine is in. For example, more users can be assigned to the machine than were initially assigned when creating the machine object, this is done with the `Add-BrokerUser` cmdlet.

For a full list of the machine configuration options available, see the `Set-BrokerMachine` command.

## MAINTENANCE MODE

There are times when it is necessary to disable machines. This can be achieved by setting the `InMaintenanceMode` property to `$true`. This puts the machine into maintenance mode. With single-session machines, this means that the broker excludes the machine from brokering decisions and does not start new sessions on them. Existing sessions are unaffected. For multi-session desktops in maintenance mode, reconnections to existing sessions are allowed, but no new sessions are created on the machine.

Machines in maintenance mode are also excluded from automatic power management, although explicit power actions are still performed.

## SEE ALSO

[about\\_Broker\\_PowerManagement](#)

[about\\_Broker\\_Desktops](#)

[New-BrokerMachine](#)

[Add-BrokerMachine](#)

[Add-BrokerMachinesToDesktopGroup](#)

[Remove-BrokerMachine](#)

[Get-BrokerMachine](#)

[Set-BrokerMachine](#)

# about\_Broker\_Policies

Jan 04, 2017

## TOPIC

Citrix Broker SDK - Access, Entitlement, and Assignment Policies

## SHORT DESCRIPTION

Overview of the site policies that control users' access to desktop and application sessions.

## LONG DESCRIPTION

For an end user to access a desktop or application resource within a site, they must have both an entitlement to use the resource, and have access to the desktop group that contains the resource.

Entitlements to use resources can be granted by one of the following means:

- o The site entitlement policy grants entitlements to launch a shared desktop or application session from a pool of shared machines.
- o The site assignment policy grants entitlements for "self service" permanent assignment of machines to users for running desktop or application sessions, and is referred to as "Assign On First Use" (AOFU)
- o Machines can be permanently assigned ("pre-assigned") to users by the administrator to run either desktop or application sessions.
- o Machines can be configured to allow automatic permanent assignment to their normal user (using the RemotePC feature).

A user must also be granted access to the desktop group that contains the resource. These access rights are controlled by the site's access policy.

The access policy controls access using details of the user's device such as whether it's connected over a local area network (LAN) or connected through Access Gateway, the user device's name, IP address or subnet, and the requested connection protocol. The user's identity can also feed into the access check allowing, for example, certain users access to resources only when locally connected to the site, but others full remote access.

Access and entitlements can be combined to allow rich and fine-grained control over which users have access to site resource from any given user device or location.

Each site has a single access policy, entitlement policy, and assignment policy. Each policy comprises a set of rules. Policies are defined by adding, removing, or changing rules.

Each site policy can also be viewed as a set of distinct policies each relating to a single desktop group. In general a group has one or more policy rules that relate to it, however each rule relates to only a single group. Thus the rules that grant entitlement and access rights to a desktop group define the policy for that group and that group only; changing this policy has no impact on the entitlement and access rights for any other other group in the site.

For detailed information about defining policy rules, see:

help New-BrokerAccessPolicyRule  
help New-BrokerEntitlementPolicyRule  
help New-BrokerAssignmentPolicyRule  
help New-BrokerAppEntitlementPolicyRule  
help New-BrokerAppAssignmentPolicyRule

The mapping of policies to the resources that they make available within a site is described briefly below. For specific information on configuring each category of resource, consult the more detailed help topics listed.

## SHARED DESKTOP AND APPLICATION SESSIONS

To grant access to a group of shared machines, use the access and entitlement policies:

- o The access policy grants access to the desktop group containing the machines to be shared.
- o The entitlement policy grants an entitlement to use one or more machines in the group to specified users or groups of users.

Groups of shared machines can be used to deliver full desktop or seamless application sessions, or both.

For more detailed information about configuring shared machines, see:

help about\_Broker\_AccessPolicy  
help about\_Broker\_EntitlementPolicy

## PRE-ASSIGNED PRIVATE MACHINES

To grant access to private machines, use the access policy and a machine assignment:

- o The access policy grants access to the desktop group containing the machines.
- o The assignment links the desktop to a specified user. You can assign a machine to just one user, multiple users or user groups. However, for single-session machines, only one user can access the machine at a time.

Private machines can be used to deliver full desktop or seamless application sessions (but not both).

For more detailed information about configuring private machines, see:

help about\_Broker\_AccessPolicy  
help Add-BrokerUser



## ASSIGN-ON-FIRST-USE (AOFU) MACHINES

To grant access to a desktop group containing assignable machines, use the access policy and the assignment policy:

- o The access policy grants access to the desktop group containing the pool of machines.
- o The assignment policy grants users a self-service entitlement to pick one or more machines from the pool.

AOFU machines can be used to deliver full desktop or seamless application sessions (but not both from the same desktop group).

For more detailed information about configuring AOFU desktops, see:

[help about\\_Broker\\_AccessPolicy](#)  
[help about\\_Broker\\_AssignmentPolicy](#)

## REMOTE PC MACHINES

The RemotePC feature allows existing physical machines to be assigned automatically to their normal user thus allowing them remote access to their own machine but without the need for the administrator to individually configure access to each machine.

For more detailed information about configuring the Remote PC feature, see:

[help about\\_Broker\\_RemotePC](#)

## SEE ALSO

[about\\_Broker\\_AccessPolicy](#)

[about\\_Broker\\_EntitlementPolicy](#)

[about\\_Broker\\_AssignmentPolicy](#)

[about\\_Broker\\_RemotePC](#)

[New-BrokerAccessPolicyRule](#)

[New-BrokerEntitlementPolicyRule](#)

[New-BrokerAssignmentPolicyRule](#)

[New-BrokerAppEntitlementPolicyRule](#)

[New-BrokerAppAssignmentPolicyRule](#)



# about\_Broker\_PostInstallPreConfiguration

Jan 04, 2017

## TOPIC

Citrix Broker SDK - Post-Installation Configuration

## SHORT DESCRIPTION

Describes how to configure the Citrix Broker Service port numbers, URL reservations, and Windows Firewall exclusions.

## LONG DESCRIPTION

The XenDesktop installer configures the Citrix Broker Service with information specified during the installation. To change that configuration, use the BrokerService.exe command-line tool on each controller you want to change.

The default installation location of BrokerService.exe is:

```
%ProgramFiles%\Citrix\Broker\Service\BrokerService.exe
```

BrokerService.exe supports the following optional command-line parameters:

-SdkPort <port> (default 80)

Configures the port on which the broker listens for requests from SDK cmdlets. If you change this default value, specify the new value in the AdminAddress parameter on broker cmdlets. For example, if you changed the port to 8080, specify it as follows:

```
Get-BrokerSite -AdminAddress localhost:8080
```

-VdaPort <port> (default 80)

Configures the port on which the broker listens for registration requests from broker machines.

-WiPort <port> (default 80)

Configures the port on which the broker listens for XML requests from StoreFront/Web Interface.

-WiSslPort <port> (default 443)

Configures the port on which the broker listens for SSL (Secure Socket Layer) XML requests from StoreFront/Web Interface.

-ConfigureFirewall

Configures Windows Firewall exclusions for the specified ports.

-Show

Shows the current configuration settings.

-Uninstall

Removes configuration settings, including Windows Firewall exclusions and URL reservations.

-LogFile <fileName>

Configures the file location for logging to a text file. The directory containing the log file must grant write access to the NetworkService account.

-Upgrade

Performs configurations required after an upgrade.

-Quiet

Suppresses console output for status messages.

-? or -Help

Shows usage information for the command.

# about\_Broker\_PowerManagement

Jan 04, 2017

## TOPIC

Citrix Broker SDK - Machine Power Management

## SHORT DESCRIPTION

Describes power management of machines used for desktops and applications.

## LONG DESCRIPTION

The Citrix Broker Service is in day-to-day control of the power state of the configured desktop and application machines. The Broker Service can control several hypervisors, each hypervisor connection being handled by its own site service, so all Broker Service communication to the hypervisor is through one of the controllers in the site.

## HYPERVISOR CONNECTIONS

Each hypervisor, or pool of linked hypervisors, is described and configured through the XdHyp pseudo-drive and associated Hyp PowerShell commands (cmdlets) which are provided by the host service snap-in. When you have first created the hypervisor connection using the host service cmdlets, you can create a broker equivalent object that references the Hyp instance using a GUID value. Use the broker's HypervisorConnection object to nominate a preferred controller for direct communication with the hypervisor on behalf of all other controllers for day-to-day power actions and status requests.

## POWER ACTIONS AND THROTTLING

The site, through the Broker Service, can control the power state of the machines used by the site for desktops and applications. Power state changes can have a number of causes:

- o Power policy rules, such as requests to shut down or suspend machines when user sessions on those machines end or are disconnected
- o If allowed, user-driven desktop restarts
- o Session launch requests requiring machines to be started
- o Pool size management, which controls the number of running machines
- o Direct administrator request using the SDK or Citrix Studio
- o Reboot schedules and cycles
- o Performing personal vDisk inventory activities
- o Cleaning machines back to the golden master image state after they have been used

The power state changes of machines hosting desktops and applications are controlled using a queuing mechanism. Actions to change the power state are assigned a priority and are sent to the hypervisor according to a throttling mechanism. This avoids overloading the hypervisor.

The queuing and throttling mechanisms take place on a per-hypervisor-connection basis; each hypervisor connection's queue is dealt with independently. You can view the contents of the queues using the Get-BrokerHostingPowerAction cmdlet. This includes recently completed, in-progress, and pending actions (that is, those due to be sent to the hypervisor

based on the throttling settings).

Each power action object comprises:

- o The machine to be acted on (Name, DNS Name, Hosting Name)
- o The action to be performed (TurnOn, TurnOff, Shutdown, Reset, Restart, Suspend, or Resume)
- o The action's priority (Base, the original priority, and Actual, the current priority)
- o The action's state (Pending, Started, Completed, Failed, Canceled, Deleted, or Lost)
- o Time stamps of the action's lifecycle points (when it was created, started, or completed)
- o Any reason the action failed

The throttling of power actions is controlled by three metadata values on the Hyp hypervisor connection object when accessed through the XdHyp pseudo-drive. The four values throttle power actions according to:

- o The maximum absolute number of in-progress power actions
- o The maximum number of in-progress power actions expressed as a percentage of the total number of machines controlled by the hypervisor connection
- o The maximum number of new power actions sent to the hypervisor per minute
- o The maximum number of in-progress PvD inventory activities expressed as a percentage of the total number of machines controlled by the hypervisor connection

You add power actions to the queue using the SDK's New-BrokerHostingPowerAction cmdlet. You cancel power actions in the queue using the Remove-BrokerHostingPowerAction cmdlet. You boost or reduce their priority using the Set-BrokerHostingPowerAction cmdlet.

You can also schedule power actions to be executed in the future, using the New-BrokerDelayedHostingPowerAction command. Only Shutdown and Suspend actions can be scheduled in this way. You can view these delayed power actions using Get-BrokerDelayedHostingPowerAction and cancel them with Remove-BrokerDelayedHostingPowerAction. When a delayed power action is ready to be executed it is deleted, and a corresponding normal power action is placed in the queue described above.

## POWER POLICY

Policy rules associated with a desktop group allow you to change power states at configurable times after session state changes, typically a set number of minutes after session disconnection or session logout.

Note that these policy rules are defined directly as properties of the desktop group.

These policy actions allow the following operations to be specified:

- o A power action to be performed at a defined period after a session is disconnected
- o A power action to be performed at a defined extended period after a session is disconnected
- o A power action to be performed at a defined period after a session is logged off

The two disconnect policy actions are designed to allow multi-stage policies such as initially suspending a machine shortly after a session disconnect occurs, and then later powering-off the machine if the session has not been reconnected.

At the set time after the session state change, the required action is added to the power action queue, and this is then throttled and processed as normal.

## POOL SIZE MANAGEMENT

You can manage flexibly the number of machines running desktops and applications using the pool size. For any given hour of the day and day of the week, this is an absolute number of machines or the percentage of the total number of machines in the desktop group. The pool size specifies the total number of machines that are always running, regardless of whether they are in use or idle. (Note: The number of machines does not depend on their idle status, but this does affect the buffer size value, which is also used to manage pool sizes.)

To start or shut down desktop machines to achieve the desired pool size, the system places power actions in the queue. Standard throttling queue management sends these to the hypervisors. A single desktop group (and its pool) can span multiple hypervisors, so actions to start and shut down machines can be added to multiple queues.

## POWER TIME SCHEMES

Each single-session desktop group can be associated with one or more power time schemes, each scheme covering a number of days of the week. The time schemes specify, for each hour of the day, whether that hour is peak or off-peak. They also specify the number of running unassigned machines maintained by the broker.

You can configure other settings, such as buffer size and any power policy rules differently for peak and off-peak hours. You can define the number of running machines, idle or in use, either as an absolute value or as a percentage of the desktop group size. Machines running desktops and applications are started (or shut down when not in use) to match the required pool size.

Each power time scheme comprises:

- o The name of the scheme
- o The pattern of days of the week covered by the scheme
- o The set of hours considered peak and off peak
- o The set of pool size values (one for each hour of the day)

The hours of the day used by time schemes are the hours in the time zone for the desktop group the scheme is associated with. You cannot associate one desktop group with multiple time schemes covering the same day of the week.

## BUFFER SIZE

In addition to the pool size, you can optionally configure two buffer sizes for each desktop group, one for peak hours and one for off-peak hours. The buffer size defines the minimum number of idle unassigned machines maintained by the broker and is specified as a percentage of the total machines in the group. These are running machines that are not used by any user session. The buffer size on its own never causes machines to shut down. It causes them to start up so a minimum number of idle machine is always available. The buffer size in conjunction with the pool size can cause machines running desktops or applications to be shut down.

## POWER MANAGEMENT OF ASSIGNED MACHINES

Automatic power management for private desktop groups provides the ability to power on all assigned machines at the transition to a peak period and respectively power off all machines at the transition to an off-peak period.

If a machine is shut down during peak hours it will not be automatically powered on again, unless the `AutomaticPowerOnForAssignedDuringPeak` property on the desktop group is also enabled.

Note that all power management facilities apply only to single session machines.

## REBOOT SCHEDULES

Reboot schedules are commonly used after image updates or to perform regular reboots of all machines in a desktop group or catalog to clear down problems resulting from a corrupt state or hung/faulty applications.

Reboot schedules allow distributing the reboot operation of all machines over a provided duration. Individual machine reboots are scheduled in a way that attempts to maintain maximum availability of machines in the group as the reboots occur, and avoid boot storms that overload the underlying infrastructure.

Reboot schedules are the only form of automatic power management that can shut down a machine while users are logged on; however the administrator can provide a warning message to be displayed to end users at a specified period prior to the shutdown taking effect.

## REBOOT CYCLES

Reboot cycles describe the dynamic execution of desktop group or catalog reboot operations. Reboot cycles can be created due to reboot schedules, or by on-demand reboot operations requested via the SDK.

`RebootCycle` objects encapsulate the details of the associated reboot operation and can be queried to show the current status.

## STATUS

You can view the status of the hypervisor connection on the broker hypervisor connection object. You can obtain any hypervisor alerts using the `Get-BrokerHypervisorAlert` cmdlet. You can check the power state of machines running desktops or applications using the relevant `Machine` or `Desktop` objects.

## SEE ALSO

[about\\_Broker\\_Concepts](#)

[about\\_Broker\\_Machines](#)

[about\\_HypHostSnapin](#)

[New-BrokerHypervisorConnection](#)



Get-BrokerHypervisorConnection  
Set-BrokerHypervisorConnection  
Remove-BrokerHypervisorConnection  
New-BrokerHostingPowerAction  
Get-BrokerHostingPowerAction  
Set-BrokerHostingPowerAction  
Remove-BrokerHostingPowerAction  
New-BrokerDelayedHostingPowerAction  
Get-BrokerDelayedHostingPowerAction  
Remove-BrokerDelayedHostingPowerAction  
New-BrokerPowerTimeScheme  
Get-BrokerPowerTimeScheme  
Set-BrokerPowerTimeScheme  
Rename-BrokerPowerTimeScheme  
Remove-BrokerPowerTimeScheme  
Get-BrokerRebootCycle  
Set-BrokerRebootCycleMetadata  
Start-BrokerRebootCycle  
Stop-BrokerRebootCycle  
Get-BrokerRebootSchedule  
Set-BrokerRebootSchedule  
New-BrokerRebootSchedule  
Remove-BrokerRebootSchedule  
New-BrokerDesktopGroup  
Get-BrokerDesktopGroup  
Set-BrokerDesktopGroup  
Add-HypMetadata  
Remove-HypMetadata

# about\_Broker\_RemotePC

Jan 04, 2017

## TOPIC

Citrix Broker SDK - RemotePC

## SHORT DESCRIPTION

Overview of the Remote PC feature.

## LONG DESCRIPTION

Remote PC allows automatic publishing of a user's physical desktop within a XenDesktop site, so that it can be accessed remotely. The configuration of Remote PC within the Citrix Broker service specifies the rules and relationships that allow the machine to successfully register with a controller in the site, and be made available for the user to start remote sessions.

When Remote PC is configured, there are two steps required for publishing a machine as a Remote PC desktop.

First a VDA must be installed on the machine and it must be configured such that it registers with a controller in a site.

Second, a user must log on to the machine. When the Citrix Broker service detects an active console session for a user, it assigns the user to the machine and publishes it in a Remote PC desktop group.

Remote PC configuration consists of defining relationships between:

- o Machines and catalogs
- o Catalogs and desktop groups
- o Desktop groups and assignment policy rules
- o Assignment policy rules and users

The Citrix Broker service automates the assignment of Remote PC machines to users in two stages. The first stage automatically imports matching unconfigured machines into the site:

- o Suitable machines are automatically added to a Remote PC catalog.
- o The machines are temporarily configured to be in one of the Remote PC desktop groups associated with the catalog.

When a suitable user logs on to the console of the machine, the second stage of the automatic configuration is performed:

- o The machine is configured to be in the desktop group that is appropriate for the user.
- o The machine is assigned to the user that has logged on.

- o The machine desktop is made available to the user remotely, configured to appear as the machine hostname.

Catalogs and desktop groups can be marked as participating in RemotePC automation with the 'IsRemotePC' property, but this property can only be set to true if various other properties of the catalog or desktop group are appropriate. Catalogs must be single-sessioned and contain physical machines. Desktop Groups must be single session, configured to deliver private assigned machines, delivering desktop sessions only.

Catalogs and desktop groups can have the 'IsRemotePC' property cleared to remove them from the RemotePC automation, but only when all RemotePC associations relating to them through RemotePCAccounts and catalog/group relationships have first been removed.

## MACHINES AND CATALOGS

Mappings are defined between machines and Remote PC catalogs through the RemotePCAccount cmdlets.

The machine to catalog mappings support the automated addition of machines to catalogs.

## PROPERTIES

RemotePCAccounts expose sets of included and excluded machine name filters specified in DOMAIN\MACHINE format. A MachinesIncluded or MachinesExcluded entry can include asterisk wildcards to generalize matches.

Each RemotePCAccount can specify the Distinguished Name (DN) of an AD container in addition to the machine name filters, in the RemotePCAccount Organisational Unit (OU) property, and this limits the machines that the account objects act on to those that reside at or below that container in the AD domain hierarchy. An AllowSubfolderMatches setting on the RemotePCAccount indicates whether the computer must exist directly within the container to trigger a match, or whether it can be in a child below the defined container in the AD hierarchy.

Note that the AD container component is optional and a special value of 'any' can be supplied in the OU field to permit the RemotePCAccount to automatically match regardless of the AD machine object location in the AD domain hierarchy. A match is still subject to machine name filtering.

The last component of the RemotePCAccount is the CatalogUid. This indicates which catalog the Remote PC automation should move the machine into when a match is found.

## CONSTRAINTS

The IsRemotePC setting must be enabled on catalogs before they can be specified in a RemotePCAccount.

There can be any number of RemotePCAccounts configured in the site as long as each specifies a unique OU. There can be only one RemotePCAccount with the 'any' OU.

## AUTOMATION

When a machine matching the criteria set up in a RemotePCAccount instance registers with one of the brokers in the site, it is automatically added to the catalog defined by the RemotePCAccount. This can take up to 30 seconds to happen after the machine registers.

When the machine registration occurs, the machine details can match more than one RemotePCAccount instance, but the machine can only be placed into one catalog, so one RemotePCAccount instance is chosen from the list that best matches

the machine. This choice is made according to the length in nodes of the DN of the AD container specification associated with the RemotePCAccount, so more specific child OUs override specifications for their parent OUs if both are present. The RemotePCAccount for the 'any' OU is always last and used only if no other instances match.

A Windows eventlog message is generated when an automated catalog assignment is performed.

## NOTES

The AD distinguished name for the container is checked when the RemotePCAccount is created, but if the container is subsequently moved or deleted, the site does not automatically accommodate this, and the RemotePCAccount must be changed or removed manually.

## Related Cmdlets

- o [New-BrokerRemotePCAccount](#)
- o [Get-BrokerRemotePCAccount](#)
- o [Set-BrokerRemotePCAccount](#)
- o [Remove-BrokerRemotePCAccount](#)
- o [New-BrokerCatalog \[-IsRemotePC <Boolean>\]](#)
- o [Set-BrokerCatalog \[-IsRemotePC <Boolean>\]](#)

## CATALOGS AND DESKTOP GROUPS

A Remote PC catalog may be associated with one or more Remote PC desktop groups. The catalog to desktop group associations support automated publishing of machines to users.

## USAGE

An association is formed via:

```
Add-BrokerDesktopGroup -RemotePCCatalog <Catalog> [-Priority <Int32>]
```

An association is broken via:

```
Remove-BrokerDesktopGroup -RemotePCCatalog <Catalog>
```

To find associated desktop groups:

```
Get-BrokerDesktopGroup -RemotePCCatalogUid <Int32>
```

## AUTOMATION

When a machine in a Remote PC catalog is not already assigned to a user, Remote PC automation temporarily places the

machine in one of the desktop groups associated with the catalog. The temporary placing of the machine in a group will be adjusted as needed when the machine assignment to a user is first made.

The desktop group chosen as the temporary home for a machine is according to the priority value supplied when making the association between the group and the catalog. The group with the highest priority (lowest numerical priority value) is chosen.

A Windows eventlog message is generated when an automated machine assignment is performed.

## PRIORITY

Each desktop group to catalog association has a priority value that can be specified when the association is made or defaults to lower than the lowest existing association priority (highest numerical value) or zero if no other association exists yet. The lower the priority numerical value, the higher the priority level. The priority value for the association is used to choose the temporary desktop group to place a new RemotePC machine in when it first registers. It is also used to decide which group to finally place the machine in when the user that the machine is being assigned to is appropriate for more than one of the associated desktop groups, usually because the desktop groups are using AD security groups for the user associations with the desktop groups.

The priority values for each association between a desktop group and a catalog are automatically maintained as unique for each catalog, and priorities can be adjusted up or down accordingly by the system. The last association created without a specified priority is always arranged to have the least priority (the highest numerical priority value).

## NOTES

The temporary placement of the machines in a desktop group is not re-evaluated if settings change after the automatic placement.

## RELATED CMDLETS

- o [Add-BrokerDesktopGroup -RemotePCCatalog <Catalog>](#)
- o [Remove-BrokerDesktopGroup \[-RemotePCCatalog <Catalog>\]](#)
- o [Get-BrokerDesktopGroup \[-RemotePCCatalogUid <Int32>\]](#)
- o [New-BrokerCatalog \[-IsRemotePC <Boolean>\]](#)
- o [Set-BrokerCatalog \[-IsRemotePC <Boolean>\]](#)
- o [New-BrokerDesktopGroup \[-IsRemotePC <Boolean>\]](#)
- o [Set-BrokerDesktopGroup \[-IsRemotePC <Boolean>\]](#)

## DESKTOP GROUPS, ASSIGNMENT POLICY RULES, AND USERS

Assignment policy rules are used to define sets of users allowed to be assigned to machines by Remote PC automation, and to determine which desktop group the machine is placed in when the user assignment is made.

## AUTOMATION

A machine is automatically assigned to a user when the Citrix Broker service sees that the user has logged on to a RemotePC machine, and the user is amongst those configured in the desktop group assignment policy rule. If this is the first assignment of a user to a machine, all the assignment policy rules for all groups associated with the machine are checked,

and the machine can be moved to a different, more appropriate desktop group if needed.

A Windows eventlog message is generated when an automated assignment of a machine to a user is made.

## MULTIPLE ASSIGNMENTS

By default, multiple automatic assignments of users to the same machine can be established if multiple users log on to the RemotePC machine, but this can be disabled if desired using a registry setting (see CTX137805).

## NOTES

User to machine assignments can still be made and removed manually through the Powershell SDK for machines in Remote PC catalogs and desktop groups.

The automatic placement of a machine in an appropriate desktop group happens when the first automatic assignment of a user to the machine is made, and this placement will not be automatically updated if the configuration subsequently changes.

Only a single assignment policy rule is allowed for each RemotePC desktop group.

## RELATED CMDLETS

- o [New-BrokerUser](#)

- o [New-BrokerAssignmentPolicyRule](#)

- o [Set-BrokerAssignmentPolicyRule](#)

## EXAMPLE

The following example creates a simple configuration that allows any user and machine in the current AD domain to participate in RemotePC.

```
# Create a Remote PC catalog
$catalog = New-BrokerCatalog -IsRemotePC $true
    -SessionSupport SingleSession
    -MachinesArePhysical $true
    -AllocationType Static
    -PersistUserChanges OnLocal
    -ProvisioningType Manual
    -Name RemotePCCatalog

# Create a Remote PC desktop group
$dg = New-BrokerDesktopGroup -IsRemotePC $true
    -SessionSupport SingleSession
    -DeliveryType DesktopsOnly
    -DesktopKind Private
    -Name RemotePCDesktopGroup

# Create an assignment policy rule for that desktop group allowing any
# domain user to match.
```

```
New-BrokerAssignmentPolicyRule -DesktopGroupUid $dg.Uid
    -IncludedUsers 'domain users'
    -Name RemotePCAPR
```

```
# Create a RemotePCAccount matching any unconfigured machine, causing the
# machines to be added to the catalog by Remote PC automation.
```

```
New-BrokerRemotePCAccount -OU 'any'
    -CatalogUid $catalog.Uid
```

```
# Associate the desktop group and catalog to permit domain users to be
# automatically assigned to machines in that catalog
```

```
Add-BrokerDesktopGroup $dg -RemotePCCatalog $catalog
```

```
#Create an access policy rule, allowing access to the users to the
#Remote PC desktop group.
```

```
New-BrokerAccessPolicyRule -IncludedUsers 'domain users'
    -DesktopGroupUid $dg.Uid
    -IncludedUserFilterEnabled $true
    -Name RemotePCAccessPolicyRule
```

# Add-BrokerApplication

Jan 04, 2017

Adds applications to a desktop group.

## Syntax

```
Add-BrokerApplication [-InputObject] <Application[]> [-DesktopGroup <DesktopGroup>] [-Priority <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-BrokerApplication [-Name] <String> [-DesktopGroup <DesktopGroup>] [-Priority <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Add-BrokerApplication cmdlet is used to associate one or more applications with an existing desktop group.

There are two parameter sets for this cmdlet, allowing you to specify the application either by its BrowserName or by an array of object references. Uids can also be substituted for the object references.

See about\_Broker\_Desktops and about\_Broker\_Applications for more information.

## Related topics

[New-BrokerApplication](#)

[Add-BrokerApplication](#)

[Add-BrokerTag](#)

[Remove-BrokerApplication](#)

[Rename-BrokerApplication](#)

[Move-BrokerApplication](#)

[Set-BrokerApplication](#)

## Parameters

**-InputObject** <Application[]>

Specifies the application to associate. Its Uid can also be substituted for the object reference.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |



**-Name**<String>

Specifies the name of the application to be associated with the desktop group.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopGroup**<DesktopGroup>

Specifies which desktop group this application should be associated with. Note that applications can only be associated with desktop groups of the AppsOnly or DesktopsAndApps delivery type.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Priority**<Int32>

Specifies the priority of the mapping between the application and desktop group where lower numbers imply higher priority with zero being highest.

If one association has a higher priority than the other, machines from that group will be selected for launching sessions until all machines are at maximum load, in maintenance mode, unregistered, or unavailable for any other reason. Only when all machines from the higher-priority group are unavailable will new connections be routed to the next lowest priority group.

If multiple associations have equal priority, load balancing does not occur among the desktop groups in these associations. Instead, the broker chooses one of these groups as the preferred group and machines from this group will be selected for launching sessions until all machines are at maximum load, in maintenance mode, unregistered, or unavailable for any other reason. Only when all machines from the preferred group are unavailable will new connections be routed to another one of these groups, which the broker chooses as next-most preferred.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 0                     |
| Accept Pipeline Input? | true (ByPropertyName) |

### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.Application, or as appropriate by property name You can pipe the application to be added to Add-BrokerApplication. You can also pipe some of the other parameters by name.

### Return Values

None

### Examples

#### ----- EXAMPLE 1 -----

```
C:\PS> Add-BrokerApplication -BrowserName "Notepad" -DesktopGroup "Private DesktopGroup"
```

Adds the application with a BrowserName of "Notepad" to the desktop group called "Private DesktopGroup".

# Add-BrokerDesktopGroup

Jan 04, 2017

Associate Remote PC desktop groups with the specified Remote PC catalog.

## Syntax

```
Add-BrokerDesktopGroup [-InputObject] <DesktopGroup[]> [-RemotePCCatalog <Catalog>] [-Priority <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-BrokerDesktopGroup [-Name] <String> [-RemotePCCatalog <Catalog>] [-Priority <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet forms relationships between Remote PC desktop groups and catalogs.

The Remote PC relationships are used by Remote PC automation to determine which desktop groups a machine in a particular Remote PC catalog can be published to. The assignment policy rules belonging to those desktop groups also determines the set of users that are allowed to be assigned to machines from the catalog.

## Related topics

[Remove-BrokerDesktopGroup](#)

[Add-BrokerCatalog](#)

[Remove-BrokerCatalog](#)

## Parameters

**-InputObject**<DesktopGroup[]>

Specifies one or more Remote PC desktop groups to add to a Remote PC catalog.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the Remote PC desktop groups to add to a Remote PC catalog based on their name properties.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-RemotePCCatalog**<Catalog>

The Remote PC catalog which the desktop groups are to be added to. Specified by name, Uid or instance.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Priority**<Int32>

Desktop group to catalog associations carry a priority number, where numerically lower values indicate a higher priority.

The priority relative to other associations determines which desktop group Remote PC automation will move a qualifying unconfigured machine into when it registers. Priority also determines which desktop group a machine will be published to when a user is assigned to the machine by Remote PC automation.

If a value is not supplied, then the desktop group association is automatically assigned a lower priority than any existing associations.

If a priority value is specified that conflicts with an existing association's priority value, then the new association is inserted with that value and existing associations are renumbered upwards to accommodate it.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | See description       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.DesktopGroup The set of Remote PC desktop groups to be added to the catalog can be piped into this cmdlet.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerDesktopGroup -IsRemotePC $true | Add-BrokerDesktopGroup -RemotePCCatalog 42  
Add all Remote PC desktop groups to Remote PC catalog 42.
```

----- **EXAMPLE 2** -----

```
C:\PS> Add-BrokerDesktopGroup -Name *MyGroup* -RemotePCCatalog RPCCat  
Add desktop groups with names containing MyGroup to Remote PC catalog with name "RPCCat".
```

# Add-BrokerMachine

Jan 04, 2017

Adds one or more machines to a desktop group.

## Syntax

```
Add-BrokerMachine [-InputObject] <Machine[]> [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-BrokerMachine [-MachineName] <String> [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Add-BrokerMachine cmdlet adds specified machines to a desktop group. There are three forms:

- o Use the -InputObject parameter to add a single machine instance or array of instances to the group.
- o Use the -MachineName parameter to add a single, named machine to the group.
- o Use pipelining to pipe machines instances to the command.

The desktop group to which the machines are added can be specified by name, unique identifier (UID), or instance.

For a machine to be used in a site, the machine must be added to a desktop group. The machine and desktop group must be compatible in order for the process to succeed; for example a machine in a single-session catalog cannot be added to a multi-session desktop group.

For more information about machines, see about\_Broker\_Machines.

## Related topics

[Add-BrokerMachinesToDesktopGroup](#)

[Remove-BrokerMachine](#)

[Get-BrokerMachine](#)

## Parameters

**-InputObject** <Machine[]>

An array of machines to add to the group.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-MachineName**<String>

The name of the single machine to add (must match the MachineName property of the machine).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopGroup**<DesktopGroup>

The desktop group to which the machines are added, specified by name, Uid, or instance.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

---

## Input Type

Citrix.Broker.Admin.SDK.Machine You can pipe in the machines you want to add.

## Return Values

None

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Add-BrokerMachine -InputObject $machine -DesktopGroup $desktopGroup
```

```
C:\PS> Add-BrokerMachine -InputObject $machine -DesktopGroup 2
```

```
C:\PS> Add-BrokerMachine $machine -DesktopGroup "MyDesktopGroup"
```

These examples all add a single machine instance to a desktop group, identifying the group by instance, UID, or name.

### ----- EXAMPLE 2 -----

```
C:\PS> Add-BrokerMachine -MachineName "MyDomain\MyMachine" -DesktopGroup 2
```

```
C:\PS> Add-BrokerMachine "MyDomain\MyMachine" -DesktopGroup "MyDesktopGroup"
```

```
C:\PS> Add-BrokerMachine "MyDomain\MyMachine" -DesktopGroup $desktopGroup
```

These examples add the machine called MyMachine to a desktop group.

### ----- EXAMPLE 3 -----

```
C:\PS> Get-BrokerMachine -Uid 3 | Add-BrokerMachine -DesktopGroup 2
```

```
C:\PS> Get-BrokerMachine -CatalogUid 4 | Add-BrokerMachine -DesktopGroup 2
```

These examples find specific machines and add them to a desktop group.



# Add-BrokerMachineConfiguration

Jan 04, 2017

Adds a machine configuration to a desktop group.

## Syntax

```
Add-BrokerMachineConfiguration [-InputObject] <MachineConfiguration[]> [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-BrokerMachineConfiguration [-Name] <String> [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Associates a machine configuration with a desktop group. The settings in the machine configuration are then applied to the machines in the desktop group.

## Related topics

[New-BrokerMachineConfiguration](#)

[Set-BrokerMachineConfiguration](#)

[Rename-BrokerMachineConfiguration](#)

[Remove-BrokerMachineConfiguration](#)

## Parameters

**-InputObject**<MachineConfiguration[]>

Machine configuration to add to the desktop group.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Name of a machine configuration to add to the desktop group.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-DesktopGroup**<DesktopGroup>

The desktop group to which the machine configurations are added, specified by name, Uid, or instance.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.MachineConfiguration The machine configuration to add to the desktop group.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

```
Add-BrokerMachineConfiguration -Name UPM\Conf1 -DesktopGroup 1
```

Adds the machine configuration named UPM\Conf1 to the desktop group with Uid 1.

----- **EXAMPLE 2** -----

```
$mc | Add-BrokerMachineConfiguration -DesktopGroup AdminDesktops
```

Adds the machine configuration \$mc to the desktop group named "AdminDesktops".

# Add-BrokerMachinesToDesktopGroup

Jan 04, 2017

Adds machines from a catalog to a desktop group.

## Syntax

```
Add-BrokerMachinesToDesktopGroup [-Catalog] <Catalog> [-DesktopGroup] <DesktopGroup> [-Count] <Int32> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Add-BrokerMachinesToDesktopGroup cmdlet adds a specified number of machines from a catalog to a desktop group.

The cmdlet adds as many machines as possible from the catalog to the desktop group, up to the specified number. The number of machines successfully added to the desktop group is returned.

The machines are added randomly from the catalog and are selected from those that are not already members of a desktop group, and not already assigned to a client, IP address, or user.

Both the catalog and desktop group can be referenced either by instance, name, or unique identifier (Uid). The allocation type of the catalog must be compatible with the type of desktop group. This means the session support (single/multi) and the allocation type (private/shared) of the catalog must match the session support and allocation type in the desktop group.

## Related topics

[Add-BrokerMachine](#)

[Remove-BrokerMachine](#)

## Parameters

### **-Catalog**<Catalog>

The catalog from which the machines are taken.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

### **-DesktopGroup**<DesktopGroup>

The desktop group to which the machines are added.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Count**<Int32>

The number of machines to add to the desktop group.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.Catalog, or the name or Uid of the catalog. You can pipe in the catalog from which the machines are taken. Alternatively, you can pipe the name or the Uid of the catalog.

**Return Values**

System.Int32

The number of machines added to the desktop group.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Add-BrokerMachinesToDesktopGroup -Catalog $catalog -DesktopGroup $desktopGroup -Count 1000
```

```
C:\PS> Add-BrokerMachinesToDesktopGroup -Catalog "MyCatalog" -DesktopGroup "MyDesktopGroup" -Count 1000
```

```
C:\PS> Add-BrokerMachinesToDesktopGroup -Catalog 23 -DesktopGroup 4 -Count 1000
```

All these examples request that a thousand machines from a catalog are added to a desktop group. The first example references both catalog and desktop group by instance. The second example references both catalog and desktop group by name. The third example references both catalog and desktop group by Uid.

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerCatalog -ProvisioningType Manual | Add-BrokerMachinesToDesktopGroup -DesktopGroup $desktopGroup -Count 10
```

This example takes ten machines from each manually provisioned catalog and adds them to the specified desktop group.

# Add-BrokerScope

Jan 04, 2017

Add the specified catalog/desktop group to the given scope(s).

## Syntax

```
Add-BrokerScope [-InputObject] <Scope[]> [-Catalog <Catalog>] [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Add-BrokerScope cmdlet is used to associate a catalog or desktop group object with given scope(s).

To add a catalog/desktop group to a scope you need permission to change the scopes of the catalog/desktop group and permission to add objects to all of the scopes you have specified.

If the catalog/desktop group is already in any scope supplied, that scope will be silently ignored.

## Related topics

### Parameters

**-InputObject**<Scope[]>

Specifies the scope(s) to add the object to. Each can take the form of either the string form of the scope's GUID or its name.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Catalog**<Catalog>

Specifies the catalog object to be added. This can take the form of an existing catalog object, a catalog Uid or name.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-DesktopGroup**<DesktopGroup>

Specifies the desktop group object to be added. This can take the form of an existing desktop group object, a desktop

group Uid or name.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

#### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

#### Input Type

Citrix.Broker.Admin.SDK.Scope You can pipe scopes to Add-BrokerScope.

#### Return Values

NONE

#### Examples

##### ----- EXAMPLE 1 -----

```
C:\PS> Add-BrokerScope -Scope Chalfont -DesktopGroup 27  
Adds the desktop group with Uid 27 to the Chalfont scope.
```



----- **EXAMPLE 2** -----

```
C:\PS> Add-BrokerScope BFC74867-C6EF-482C-996F-3E0D340E96AC -Catalog BangaloreMachines  
Adds the BangaloreMachines catalog to the scope with the specified ScopeID.
```

# Add-BrokerTag

Jan 04, 2017

Associate a tag with another object.

## Syntax

```
Add-BrokerTag [-InputObject] <Tag[]> [-Desktop <Desktop>] [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-BrokerTag [-Name] <String> [-Desktop <Desktop>] [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Associates one or more tags with another object.

## Related topics

[Get-BrokerTag](#)

[New-BrokerTag](#)

[Remove-BrokerTag](#)

[Rename-BrokerTag](#)

## Parameters

**-InputObject**<Tag[]>

Specifies one or more tag objects.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies a tag by name.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Desktop**<Desktop>

Associates the tag with a desktop.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-DesktopGroup**<DesktopGroup>

Associates the tag with a desktop group.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|               |                                                                                        |
|---------------|----------------------------------------------------------------------------------------|
| Required?     | false                                                                                  |
| Default Value | localhost. Once a value is provided by any cmdlet, this value will become the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Input Type

Citrix.Broker.Admin.SDK.Tag Tags may be specified through pipeline input.

## Return Values

None

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $desktop = Get-BrokerDesktop -Uid 1
C:\PS> Add-BrokerTag -Name 'Tag1' -Desktop $desktop
Associates 'Tag1' with Desktop $desktop.
```

### ----- EXAMPLE 2 -----

```
C:\PS> $desktop = Get-BrokerDesktop -Uid 1
C:\PS> New-BrokerTag 'Tag2' | Add-BrokerTag -Desktop $desktop
Creates a new tag with name 'Tag2' and associates it with Desktop $desktop.
```

# Add-BrokerUser

Jan 04, 2017

Creates an association between a user and another broker object

## Syntax

```
Add-BrokerUser [-InputObject] <User[]> [-Application <Application>] [-SessionLinger <SessionLinger>] [-SessionPreLaunch <SessionPreLaunch>] [-Machine <Machine>] [-PrivateDesktop <PrivateDesktop>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-BrokerUser [-Name] <String> [-Application <Application>] [-SessionLinger <SessionLinger>] [-SessionPreLaunch <SessionPreLaunch>] [-Machine <Machine>] [-PrivateDesktop <PrivateDesktop>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Add-BrokerUser cmdlet adds broker user objects to another specified object, such as a broker private desktop. This depends on the target object type:

- o Machine - assign the broker machine to the specified user(s); when the machine is subsequently added to a desktop group, the desktop is also assigned to the same user(s).
- o PrivateDesktop - assign the desktop to the specified user(s).
- o Application - assign the application to the specified user(s).

## Related topics

[Get-BrokerUser](#)

[Remove-BrokerUser](#)

## Parameters

**-InputObject**<User[]>

The user objects to add.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The name of the user or users to be added.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Application**<Application>

The application to which the user is to be associated.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-SessionLinger**<SessionLinger>

The session linger setting to which the user is to be assigned.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-SessionPreLaunch**<SessionPreLaunch>

The session pre-launch setting to which the user is to be assigned.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-Machine**<Machine>

The machine to which the user is to be assigned

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                |
|------------------------|----------------|
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-PrivateDesktop**<PrivateDesktop>

The desktop to which the user is to be assigned

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.USer You can pipe the users to be added to Add-BrokerUser.

Return Values

None

Notes

Specify one of the -Machine or -PrivateDesktop or -Application parameters only.

Examples

----- **EXAMPLE 1** -----

```
Add-BrokerUser "DOMAIN\UserName" -PrivateDesktop "DOMAIN\MachineName"
```

Assign the specified private desktop to the specified user.

----- **EXAMPLE 2** -----

```
Add-BrokerUser "DOMAIN\UserName" -Application "ApplicationName"
```

Assign the specified application to the specified user.

----- **EXAMPLE 3** -----

```
Add-BrokerUser "DOMAIN\UserName" -Application "FolderName"\ApplicationName"
```

Assign the specified application, in the specified folder, to the specified user.



# Disconnect-BrokerSession

Jan 04, 2017

Disconnect a session.

## Syntax

```
Disconnect-BrokerSession [-InputObject] <Session[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Disconnects the specified session.

If the session is active, no warning is issued to the user before that session is disconnected.

After disconnection, sessions enter a Disconnected state. In a Disconnected state, a session still exists but there is no remote connection to that session.

## Related topics

[Get-BrokerSession](#)

[Stop-BrokerSession](#)

## Parameters

**-InputObject**<Session[]>

Identifies the session(s) to disconnect. This can be expressed as either a session Uid or a session object.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.Session The sessions to disconnect can be piped into this cmdlet.

### Return Values

None

### Notes

This operation is non-blocking and returns before it completes. The operation, however, is unlikely to fail unless there are communication problems between the controller and the machine, or if bad arguments are passed to the cmdlet itself or if the machine cannot successfully execute the operation.

The transient nature of sessions means that the list of session objects or UIDs supplied to Disconnect-BrokerSession could consist of valid and invalid sessions. Invalid sessions are detected and disregarded and the disconnect session operation is invoked on only the valid sessions.

The system can fail to disconnect the session if the machine is not in an appropriate state or if there are problems in communicating with the machine. When a disconnect is requested the system detects if the operation was initiated successfully or not by the machine. As this operation is non-blocking the system doesn't detect or report whether the disconnect ultimately succeeded or failed after it was started.

Disconnect failures are reported through the broker SDK error handling mechanism (see about\_Broker\_ErrorHandling). In the event of errors the SdkErrorRecord error status is set to SessionOperationFailed and its error data dictionary is populated with the following entries:

- o OperationsAttemptedCount: The number of operations attempted.
- o OperationsFailedCount - The number of failed operations.
- o OperationsSucceededCount - The number of successfully executed operations.
- o UnresolvedSessionFailuresCount - The number of operations that failed due to invalid sessions being supplied.
- o OperationInvocationFailuresCount - The number of operations that failed because they could not be invoked on the machine.
- o DesktopExecutionFailuresCount - The number of operations that failed because they could not be successfully executed

by the machine.

The SdkErrorRecord message will also display the number of attempted, failed and successful operations in the following format:

```
"Session operation error - attempted:<OperationsAttemptedCount>, failed:<OperationsFailedCount>, succeeded:
<OperationsSucceededCount>"
```

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerSession -UserName MyDomain\MyAccount | Disconnect-BrokerSession
Attempts to disconnect all of the sessions for the user MyDomain\MyAccount.
```

### ----- EXAMPLE 2 -----

```
C:\PS> $desktop = Get-BrokerDesktop -DNSName MyMachine.MyDomain.com
C:\PS> Disconnect-BrokerSession $desktop.SessionUid
Disconnects the session on MyMachine.
```

### ----- EXAMPLE 3 -----

```
C:\PS> trap [Citrix.Broker.Admin.SDK.SdkOperationException]
C:\PS> {
C:\PS> write $("Exception name = " + $_.Exception.GetType().FullName)
C:\PS> write $("SdkOperationException.Status = " + $_.Exception.Status)
C:\PS> write $("SdkOperationException.ErrorData=")
C:\PS> $_.Exception.ErrorData
C:\PS>
C:\PS> write $("SdkOperationException.InnerException = " + $_.Exception.InnerException)
C:\PS> $_.Exception.InnerException
C:\PS> continue
C:\PS> }
C:\PS>
C:\PS> Disconnect-BrokerSession -InputObject 10,11,12
Attempts to disconnect sessions 10, 11 and 12. Traps and displays the error information.
```

# Export-BrokerDesktopPolicy

Jan 04, 2017

Gets the site wide Citrix Group Policy settings.

## Syntax

```
Export-BrokerDesktopPolicy [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Export-BrokerDesktopPolicy returns an array of bytes containing the site-wide Citrix Group Policy settings. These policy settings are applied to every machine in the site.

## Related topics

[Import-BrokerDesktopPolicy](#)

[New-BrokerConfigurationSlot](#)

[New-BrokerMachineConfiguration](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

None

## Return Values

System.Byte[]

The configuration data as an opaque binary blob. This will be null if no site wide Citrix Group Policy settings are in place.

## Notes

Export-BrokerDesktopPolicy performs a specialized operation. Direct usage of it in scripts is discouraged, and could result in data corruption. It is recommended that this operation only be performed via the Citrix Studio.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> $policy = Export-BrokerDesktopPolicy
```

This command exports the site wide Citrix Group Policy settings.

# Get-BrokerAccessPolicyRule

Jan 04, 2017

Gets rules from the site's access policy.

## Syntax

```
Get-BrokerAccessPolicyRule [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerAccessPolicyRule [-Name] <String> [-AllowedConnections <AllowedConnection>] [-AllowedUsers <AllowedUser>] [-Description <String>] [-DesktopGroupName <String>] [-DesktopGroupUid <Int32>] [-Enabled <Boolean>] [-ExcludedClientIPFilterEnabled <Boolean>] [-ExcludedClientName <String>] [-ExcludedClientNameFilterEnabled <Boolean>] [-ExcludedSmartAccessFilterEnabled <Boolean>] [-ExcludedSmartAccessTag <String>] [-ExcludedUser <User>] [-ExcludedUserFilterEnabled <Boolean>] [-IncludedClientIPFilterEnabled <Boolean>] [-IncludedClientName <String>] [-IncludedClientNameFilterEnabled <Boolean>] [-IncludedSmartAccessFilterEnabled <Boolean>] [-IncludedSmartAccessTag <String>] [-IncludedUser <User>] [-IncludedUserFilterEnabled <Boolean>] [-Metadata <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns rules matching the specified search criteria from the site's access policy. If no search criteria are specified, all rules in the access policy are obtained.

An access policy rule defines a set of connection filters and access control rights relating to a desktop group. These allow fine-grained control of what access is granted to a desktop group based on details of, for example, a user's endpoint device, its address, and the user's identity.

----- BrokerAccessPolicyRule Object

A `BrokerAccessPolicyRule` object represents a single rule within the site's access policy. For a user to gain access to a desktop group via the rule their connection must match all its enabled include filters, and none of its enabled exclude filters. The object contains the following properties:

-- AllowedConnections (Citrix.Broker.Admin.SDK.AllowedConnection)

Controls whether connections must be local or via Access Gateway, and if so whether specified SmartAccess tags must be provided by Access Gateway with the connection. This property forms part of the included SmartAccess tags filter.

For a detailed description of this property see "help about\_Broker\_AccessPolicy".

-- AllowedProtocols (System.String[])

Protocols (for example HDX, RDP) available to the user for sessions delivered from the rule's desktop group. If the user gains access to a desktop group by multiple rules, the allowed protocol list is the combination of the protocol lists from all those rules.

If the protocol list is empty, access to the desktop group is implicitly denied.

-- AllowedUsers (Citrix.Broker.Admin.SDK.AllowedUser)

Controls the behavior of the included users filter. This can restrict access to a list of named users or groups, or allow access to any authenticated user. For a detailed description of this property see "help about\_Broker\_AccessPolicy".

-- AllowRestart (System.Boolean)

Indicates if the user can restart sessions delivered from the rule's desktop group. Session restart is handled as follows: For sessions on single-session power-managed machines, the machine is powered off, and a new session launch request made; for sessions on multi-session machines, a logoff request is issued to the session, and a new session launch request made; otherwise the property is ignored.

-- Description (System.String)

An optional description of the rule. The text is purely informational for the administrator, it is never visible to the end user.

-- DesktopGroupName (System.String)

The name of the desktop group to which the rule applies.

-- DesktopGroupUid (System.Int32)

The unique ID of the desktop group to which the rule applies.

-- Enabled (System.Boolean)

Indicates whether the rule is enabled. A disabled rule is ignored when evaluating the site's access policy.

-- ExcludedClientIPFilterEnabled (System.Boolean)

Indicates whether the excluded client IP filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- ExcludedClientIPs (Citrix.Broker.Admin.SDK.ChbIPAddressRange[])

IP addresses of user devices explicitly denied access to the rule's desktop group. Addresses can be specified as simple numeric addresses or as subnet masks (for example, 10.40.37.5 or 10.40.0.0/16). This property forms part of the excluded client IP address filter.

-- ExcludedClientNameFilterEnabled (System.Boolean)

Indicates whether the excluded client name filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- ExcludedClientNames (System.String[])

Names of user devices explicitly denied access to the rule's desktop group. This property forms part of the excluded client names filter.

-- ExcludedSmartAccessFilterEnabled (System.Boolean)

Indicates whether the excluded SmartAccess tags filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- ExcludedSmartAccessTags (System.String[])

SmartAccess tags which explicitly deny access to the rule's desktop group if any occur in those provided by Access Gateway with the user's connection. This property forms part of the excluded SmartAccess tags filter.

-- ExcludedUserFilterEnabled (System.Boolean)

Indicates whether the excluded users filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- ExcludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

Users and groups who are explicitly denied access to the rule's desktop group. This property forms part of the excluded users filter.

-- HdxSslEnabled (System.Boolean)

Indicates whether SSL encryption is enabled for sessions delivered from the rule's desktop group.

-- IncludedClientIPFilterEnabled (System.Boolean)

Indicates whether the included client IP filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- IncludedClientIPs (Citrix.Broker.Admin.SDK.ChbIPAddressRange[])

IP addresses of user devices allowed access to the rule's desktop group. Addresses can be specified as simple numeric addresses or as subnet masks (for example, 10.40.37.5 or 10.40.0.0/16). This property forms part of the included client IP address filter.

-- IncludedClientNameFilterEnabled (System.Boolean)

Indicates whether the included client names filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- IncludedClientNames (System.String[])

Names of user devices allowed access to the rule's desktop group. This property forms part of the included client names filter.

-- IncludedSmartAccessFilterEnabled (System.Boolean)

Indicates whether the included SmartAccess tags filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- IncludedSmartAccessTags (System.String[])

The SmartAccess tags which grant access to the rule's desktop group if any occur in those provided by Access Gateway with the user's connection. This property forms part of the excluded SmartAccess tags filter.

-- IncludedUserFilterEnabled (System.Boolean)

Indicates whether the included users filter is enabled. If the filter is disabled it is ignored when the rule is evaluated.

-- IncludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

Users and groups who are granted access to the rule's desktop group. This property forms part of the included users filter.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

A collection of arbitrary key/value pairs that can be associated with the rule. The administrator can use these values for any purpose; they are not used by the site itself in any way.



-- Name (System.String)

Administrative name of the rule. Each rule in the site's access policy must have a unique name.

-- Uid (System.Int32)

Unique ID of the rule itself.

Related topics

[New-BrokerAccessPolicyRule](#)

[Set-BrokerAccessPolicyRule](#)

[Rename-BrokerAccessPolicyRule](#)

[Remove-BrokerAccessPolicyRule](#)

Parameters

**-Uid**<Int32>

Gets only the rule with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Name**<String>

Gets only rules with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AllowedConnections**<AllowedConnection>

Gets only rules that have the specified value in the AllowedConnections property of their included SmartAccess tags filter.

Valid values are Filtered, NotViaAG, and ViaAG.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AllowedUsers**<AllowedUser>

Gets only rules that have the specified value in the AllowedUsers property of their included users filter.

Valid values are Filtered, AnyAuthenticated, Any, AnonymousOnly and FilteredOrAnonymous.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Gets only rules with the specified description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupName**<String>

Gets only rules applying to desktop groups with names matching the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Gets only rules that apply to the desktop group with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Accept Pipeline Input? |       |

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Gets only rules that are in the specified state, either enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedClientIPFilterEnabled**<Boolean>

Gets only rules that have their excluded client IP address filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedClientName**<String>

Gets only rules that have the specified client name in their excluded client names filter (whether the filter is enabled or not).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedClientNameFilterEnabled**<Boolean>

Gets only rules that have their excluded client name filter enabled (\$true) or disabled (\$false).

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ExcludedSmartAccessFilterEnabled**<Boolean>

Gets only rules that have their excluded SmartAccess tags filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedSmartAccessTag**<String>

Gets only rules that have the specified SmartAccess tag in their excluded SmartAccess tags filter (whether the filter is enabled or not).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUser**<User>

Gets only rules that have the specified user in their excluded users filter (whether the filter is enabled or not).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUserFilterEnabled**<Boolean>

Gets only rules that have their excluded user filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

|  |  |
|--|--|
|  |  |
|--|--|

**-IncludedClientIPFilterEnabled**<Boolean>

Gets only rules that have their included client IP address filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedClientName**<String>

Gets only rules that have the specified user device name in their included client names filter (whether the filter is enabled or not).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedClientNameFilterEnabled**<Boolean>

Gets only rules that have their included client name filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedSmartAccessFilterEnabled**<Boolean>

Gets only rules that have their included SmartAccess tags filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedSmartAccessTag**<String>

Gets only rules that have the specified SmartAccess tag in their included SmartAccess tags filter (whether the filter is enabled or not).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUser**<User>

Gets only rules that have the specified user in their included users filter (whether the filter is enabled or not).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUserFilterEnabled**<Boolean>

Gets only rules that have their included user filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None You cannot pipe input into this cmdlet.

Return Values



Citrix.Broker.Admin.SDK.AccessPolicyRule

Get-BrokerAccessPolicyRule returns all access policy rules that match the specified selection criteria.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerAccessPolicyRule
```

Returns all access policy rules. This offers a complete description of the current site's access policy.

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerAccessPolicyRule -Enabled $true -IncludedUser sales\tech-support
```

Returns all rules that are both enabled and explicitly include the SALES\tech-support group in their included users filter.

# Get-BrokerAdminFolder

Jan 04, 2017

Get the admin folders in this site.

## Syntax

```
Get-BrokerAdminFolder [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerAdminFolder [[-Name] <String>] [-DirectChildAdminFolders <Int32>] [-DirectChildApplications <Int32>] [-FolderName <String>] [-LastChangeId <Guid>] [-Metadata <String>] [-ParentAdminFolderUid <Int32>] [-TotalChildApplications <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Get-BrokerAdminFolder cmdlet gets admin folders in this site.

Without parameters, Get-BrokerAdminFolder gets all the admin folders that have been created. You can also use the parameters of Get-BrokerAdminFolder to filter the results to just the folders you're interested in. You can also identify folders by their UIDs or their FolderNames.

----- BrokerAdminFolder Object

A folder for use in the administration console for organising other objects. E.g. BrokerApplication objects

-- DirectChildAdminFolders (System.Int32)

The number of admin folders with this folder as a direct parent

-- DirectChildApplications (System.Int32)

The number of applications in this admin folder (does not include any applications in child folders)

-- FolderName (System.String)

The simple name of this folder within any parent folder

-- LastChangeId (System.Guid)

A random GUID assigned whenever there is a change anywhere in the hierarchy of admin folders below this node; each change updates this value on the changed folder and all parents all the way up to the root folder. Note that nodes below any change do not have their LastChangeId value updated

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Holds any metadata associated with the admin folder

-- Name (System.String)

The full name of the folder including the full parent hierarchy separated by back-slash characters and including a trailing back-slash

-- ParentAdminFolderUid (System.Int32)

The UID of the parent admin folder; the root folder references itself (zero)

-- TotalChildApplications (System.Int32)

The number of applications in this admin folder (including any applications in child folders)

-- Uid (System.Int32)

The unique ID of the admin folder (the root folder has the value zero)

## Related topics

[New-BrokerAdminFolder](#)

[Remove-BrokerAdminFolder](#)

## Parameters

**-Uid**<Int32>

Gets only the admin folder with the specified unique identifier.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Name**<String>

Gets admin folders matching the specified name (if no trailing backslash is supplied, it is assumed).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DirectChildAdminFolders**<Int32>

Gets admin folders with the specified number of child folders.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-DirectChildApplications<Int32>**

Gets admin folders with the specified number of applications (excluding those in sub-folders).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-FolderName<String>**

Gets only the admin folders matching the specified simple folder name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastChangeId<Guid>**

Gets only the admin folders with the specified value for LastChangeId.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata<String>**

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ParentAdminFolderUid**<Int32>

Gets only admin folders with the specified parent admin folder UID value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TotalChildApplications**<Int32>

Gets admin folders with the specified number of applications (including those in sub-folders).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | 250   |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None Input cannot be piped to this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.AdminFolder

Returns admin folders.

Examples

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerAdminFolder  
Return all administration folders.

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerAdminFolder -Uid 1  
Get the administration folder with Uid 1.

# Get-BrokerAppAssignmentPolicyRule

Jan 04, 2017

Gets application rules from the site's assignment policy.

## Syntax

```
Get-BrokerAppAssignmentPolicyRule [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerAppAssignmentPolicyRule [[-Name] <String>] [-Description <String>] [-DesktopGroupUid <Int32>] [-Enabled <Boolean>] [-ExcludedUser <User>] [-ExcludedUserFilterEnabled <Boolean>] [-IncludedUser <User>] [-IncludedUserFilterEnabled <Boolean>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns application rules matching the specified search criteria from the site's assignment policy. If no search criteria are specified, all application rules in the assignment policy are obtained.

An application rule in the assignment policy defines the users who are entitled to a self-service persistent machine assignment from the rule's desktop group; once assigned the machine can run one or more applications published from the group.

----- BrokerAppAssignmentPolicyRule Object

The BrokerAppAssignmentPolicyRule object represents a single application rule within the site's assignment policy. It contains the following properties:

-- Description (System.String)

An optional description of the rule. The text is purely informational for the administrator, it is never visible to the end user.

-- DesktopGroupUid (System.Int32)

The unique ID of the desktop group to which the rule applies.

-- Enabled (System.Boolean)

Indicates whether the rule is enabled. A disabled rule is ignored when evaluating the site's assignment policy.

-- ExcludedUserFilterEnabled (System.Boolean)

Indicates whether the excluded users filter is enabled. If the filter is disabled then any user entries in the filter are ignored when assignment policy rules are evaluated.

-- ExcludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

The excluded users filter of the rule, that is, the users and groups who are explicitly denied an entitlement to a machine assignment from the rule's desktop group.

-- IncludedUserFilterEnabled (System.Boolean)

Indicates whether the included users filter is enabled. If the filter is disabled then any user who satisfies the requirements of



the access policy is implicitly entitled to the machine assignment described by the rule.

-- IncludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

The included users filter of the rule, that is, the users and groups who are entitled to a machine assignment from the rule's desktop group.

-- Name (System.String)

The administrative name of the rule. Each rule in the site's assignment policy must have a unique name (irrespective of whether they are desktop or application rules).

-- Uid (System.Int32)

The unique ID of the rule itself.

## Related topics

[New-BrokerAppAssignmentPolicyRule](#)

[Set-BrokerAppAssignmentPolicyRule](#)

[Rename-BrokerAppAssignmentPolicyRule](#)

[Remove-BrokerAppAssignmentPolicyRule](#)

## Parameters

**-Uid**<Int32>

Gets the application rule with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Name**<String>

Gets only application rules with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Gets only application rules with the specified description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Gets only application rules that apply to the desktop group with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Gets only application rules that are in the specified state, either enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUser**<User>

Gets only application rules that have the specified user in their excluded users filter (whether the filter is enabled or not)

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUserFilterEnabled**<Boolean>

Gets only application rules that have their excluded user filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUser**<User>

Gets only application rules that have the specified user in their included users filter (whether the filter is enabled or not).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUserFilterEnabled**<Boolean>

Gets only application rules that have their included user filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.AppAssignmentPolicyRule

Get-BrokerAppAssignmentPolicyRule returns all application rules in the assignment policy that match the specified selection criteria.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerAppAssignmentPolicyRule

Returns all application rules from the assignment policy. This offers a complete description of the current site's assignment policy with respect to machine assignment entitlements for delivery of application sessions from private desktop groups.

----- **EXAMPLE 2** -----

C:\PS> \$dg = Get-BrokerDesktopGroup 'Sales Support'

C:\PS> Get-BrokerAppAssignmentPolicyRule -DesktopGroupUid \$dg.Uid

Returns the rule in the assignment policy that gives users entitlements to machine assignments in the Sales Support desktop group for delivery of application sessions.

# Get-BrokerAppEntitlementPolicyRule

Jan 04, 2017

Gets application rules from the site's entitlement policy.

## Syntax

```
Get-BrokerAppEntitlementPolicyRule [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerAppEntitlementPolicyRule [[-Name] <String>] [-Description <String>] [-DesktopGroupUid <Int32>] [-Enabled <Boolean>] [-ExcludedUser <User>] [-ExcludedUserFilterEnabled <Boolean>] [-IncludedUser <User>] [-IncludedUserFilterEnabled <Boolean>] [-SessionReconnection <SessionReconnection>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns application rules matching the specified search criteria from the site's entitlement policy. If no search criteria are specified, all application rules in the entitlement policy are obtained.

An application rule in the entitlement policy defines the users who are allowed per-session access to a machine to run one or more applications published from the rule's desktop group.

----- BrokerAppEntitlementPolicyRule Object

The BrokerAppEntitlementPolicyRule object represents a single application rule within the site's entitlement policy. It contains the following properties:

-- Description (System.String)

Optional description of the rule. The text is purely informational for the administrator, it is never visible to the end user.

-- DesktopGroupUid (System.Int32)

The unique ID of the desktop group to which the rule applies.

-- Enabled (System.Boolean)

Indicates whether the rule is enabled. A disabled rule is ignored when evaluating the site's entitlement policy.

-- ExcludedUserFilterEnabled (System.Boolean)

Indicates whether the excluded users filter of the rule is enabled. If the filter is disabled then any user entries in the filter are ignored when entitlement policy rules are evaluated.

-- ExcludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

The excluded users filter of the rule, that is, the users and groups who are explicitly denied entitlements to use published applications from the associated desktop group.

-- IncludedUserFilterEnabled (System.Boolean)

Indicates whether the included users filter of the rule is enabled. If the filter is disabled then any user who satisfies the

requirements of the access policy is implicitly granted an entitlement to an application session by the rule.

-- IncludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

The included users filter of the rule, that is, the users and groups who are granted an entitlement to an application session by the rule.

If a user appears explicitly in the excluded users filter of the rule or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

-- Name (System.String)

The administrative name of the rule. Each rule in the site's entitlement policy must have a unique name (irrespective of whether they are desktop or application rules).

-- SessionReconnection (Citrix.Broker.Admin.SDK.SessionReconnection)

Defines reconnection (roaming) behavior for sessions launched using this rule. Session reconnection control is an experimental and unsupported feature.

-- Uid (System.Int32)

The unique ID of the rule itself.

## Related topics

[New-BrokerAppEntitlementPolicyRule](#)

[Set-BrokerAppEntitlementPolicyRule](#)

[Rename-BrokerAppEntitlementPolicyRule](#)

[Remove-BrokerAppEntitlementPolicyRule](#)

## Parameters

**-Uid**<Int32>

Gets the application rule with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Name**<String>

Gets only application rules with the specified name.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Gets only application rules with the specified description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Gets only the application rule that applies to the desktop group with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Gets only application rules that are in the specified state, either enabled (\$true), or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUser**<User>

Gets only application rules that have the specified user in their excluded users filter (whether the filter is enabled or not).

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |



|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ExcludedUserFilterEnabled**<Boolean>

Gets only application rules that have their excluded user filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUser**<User>

Gets only application rules that have the specified user in their included users filter (whether the filter is enabled or not).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUserFilterEnabled**<Boolean>

Gets only application rules that have their included user filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionReconnection**<SessionReconnection>

Gets only application rules with the specified session reconnection behavior.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule

Get-BrokerAppEntitlementPolicyRule returns all application rules from the entitlement policy that match the specified selection criteria.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerAppEntitlementPolicyRule
```

Returns all application rules from the entitlement policy. This offers a complete description of the current site's entitlement policy with respect to application entitlements from shared desktop groups.

### ----- EXAMPLE 2 -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Customer Support'
```

```
C:\PS> Get-BrokerAppEntitlementPolicyRule -DesktopGroupUid $dg.Uid
```

Returns the application rule in the entitlement policy that grants users an entitlement to application sessions in the Customer Support desktop group.

# Get-BrokerApplication

Jan 04, 2017

Get the applications published on this site.

## Syntax

```
Get-BrokerApplication [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerApplication [-Name] <String> [-AdminFolderName <String>] [-AdminFolderUid <Int32>] [-ApplicationName <String>] [-ApplicationType <ApplicationType>] [-AssociatedDesktopGroupPriority <Int32>] [-AssociatedDesktopGroupUid <Int32>] [-AssociatedDesktopGroupUUID <Guid>] [-AssociatedUserFullName <String>] [-AssociatedUserName <String>] [-AssociatedUserUPN <String>] [-BrowserName <String>] [-ClientFolder <String>] [-CommandLineArguments <String>] [-CommandLineExecutable <String>] [-CpuPriorityLevel <CpuPriorityLevel>] [-Description <String>] [-Enabled <Boolean>] [-IconFromClient <Boolean>] [-IconUid <Int32>] [-MetadataKey <String>] [-Metadata <String>] [-PublishedName <String>] [-SecureCmdLineArgumentsEnabled <Boolean>] [-ShortcutAddedToDesktop <Boolean>] [-ShortcutAddedToStartMenu <Boolean>] [-StartMenuFolder <String>] [-UserFilterEnabled <Boolean>] [-UUID <Guid>] [-Visible <Boolean>] [-WaitForPrinterCreation <Boolean>] [-WorkingDirectory <String>] [-DesktopUid <Int32>] [-SessionUid <Int64>] [-UserSID <String>] [-DesktopGroupUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Get-BrokerApplication cmdlet gets the published applications that are hosted on this site.

Without parameters, Get-BrokerApplication gets all the applications that have been published, regardless of whether they are visible to users or not. You can also use the parameters of Get-BrokerApplication to filter the results to just the applications you're interested in. You can also identify applications by their UIDs or their BrowserNames.

For more information about applications, see about\_Broker\_Applications.

----- BrokerApplication Object

The BrokerApplication object represents a published application in the site. It contains the following properties:

-- AdminFolderName (System.String)

The name of the admin folder the application is in (including trailing backslash), or the empty string if the application is at the root level

-- AdminFolderUid (System.Int32)

The Uid of the admin folder the application is in (if any)

-- ApplicationName (System.String)

The simple name of the application within its parent admin folder (if any)

-- ApplicationType (Citrix.Broker.Admin.SDK.ApplicationType)

The type of the application, whether HostedOnDesktop or InstalledOnClient.

-- AssociatedDesktopGroupPriorities (System.Int32[])

List of associated desktop group priorities. Associated desktop groups is the list of desktop groups on which the application is published. When launching an application an available machine from one of the associated groups is selected. Desktop groups are searched for available machines in order of their priority.

-- AssociatedDesktopGroupUids (System.Int32[])

List of associated desktop group uids. Associated desktop groups is the list of desktop groups on which the application is published. The list is sorted by priority, with the highest priority group first.

-- AssociatedDesktopGroupUUIDs (System.Guid[])

List of associated desktop group UUIDs. Associated desktop groups is the list of desktop groups on which the application is published. The list is sorted by priority, with the highest priority group first.

-- AssociatedUserFullNames (System.String[])

List of associated users (full names). Associated users is the list of users who are given access using the application/user mapping filter.

-- AssociatedUserNames (System.String[])

List of associated users (SAM names). Associated users is the list of users who are given access using the application/user mapping filter.

-- AssociatedUserUPNs (System.String[])

List of associated users (user principle names). Associated users is the list of users who are given access using the application/user mapping filter.

-- BrowserName (System.String)

Unique browser name used to identify this application to other components in the site. This value is not visible to the end users.

-- ClientFolder (System.String)

The folder that the application belongs to as the user sees it.

-- CommandLineArguments (System.String)

The command-line arguments to use when launching the executable.

-- CommandLineExecutable (System.String)

The name including the full path of the executable file to launch.

-- CpuPriorityLevel (Citrix.Broker.Admin.SDK.CpuPriorityLevel)

The CPU priority of the launched process. Valid values are: Low, BelowNormal, Normal, AboveNormal, and High.

-- Description (System.String)

Optional application description. This description is visible to the end users.

-- Enabled (System.Boolean)

Specifies whether or not this application can be launched.

-- IconFromClient (System.Boolean)

Specifies if the app icon should be retrieved from the application on the client. This is reserved for possible future use, and all applications of type HostedOnDesktop cannot set or change this value.

-- IconUid (System.Int32?)

The icon UID used for this application. If not specified a generic icon is used.

-- MetadataKeys (System.String[])

All key names of metadata items associated with this application.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Metadata for this application.

-- Name (System.String)

Unique administrative name of application; this will include any parent admin folder hierarchy separated by backslash characters.

-- PublishedName (System.String)

Published name of application as seen by end user. If not specified value used defaults to the administrative name.

-- SecureCmdLineArgumentsEnabled (System.Boolean)

Specifies whether the command-line arguments should be secured.

-- ShortcutAddedToDesktop (System.Boolean)

Specifies whether a shortcut to the application should be placed on the user device.

-- ShortcutAddedToStartMenu (System.Boolean)

Specifies whether a shortcut to the application should be placed in the user's Start menu on their user device.

-- StartMenuFolder (System.String)

The name of the Start menu folder that holds the application shortcut.

-- Uid (System.Int32)

A unique identifier of an application.

-- UserFilterEnabled (System.Boolean)

Indicates if application-specific user filter is enabled.

-- UUID (System.Guid)

UUID of the application.

-- Visible (System.Boolean)

Specifies if the application is visible to users.

-- WaitForPrinterCreation (System.Boolean)

Specifies whether the VDA delays starting the app until printers are set up or not.

-- WorkingDirectory (System.String)

The working directory the executable is launched from.

### Related topics

[New-BrokerApplication](#)

[Add-BrokerApplication](#)

[Remove-BrokerApplication](#)

[Rename-BrokerApplication](#)

[Move-BrokerApplication](#)

[Set-BrokerApplication](#)

### Parameters

**-UId**<Int32>

Gets only the application with the specified unique identifier.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Name**<String>

Gets only the applications matching the specified name (including any parent admin folder hierarchy).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-AdminFolderName**<String>

Gets applications that are in admin folders matching the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminFolderUid**<Int32>

Gets applications that are in the specified admin folder.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationName**<String>

Gets applications that match the specified simple name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationType**<ApplicationType>

Gets applications that match the type specified: HostedOnDesktop or InstalledOnClient.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedDesktopGroupPriority**<Int32>

Gets applications with an associated desktop group identified by priority assigned to the pairing between an application and desktop group.

Associated desktop group is a desktop group on which the application is published.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedDesktopGroupUid<Int32>**

Gets applications with an associated desktop group identified by the desktop group UID.

Associated desktop group is a desktop group on which the application is published.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedDesktopGroupUUID<Guid>**

Gets applications with an associated desktop group identified by the desktop group UUID.

Associated desktop group is a desktop group on which the application is published.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserFullName<String>**

Gets applications with an associated user identified by their full name (usually 'first-name last-name').

If the 'UserFilterEnabled' property is true then access to the application is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserName**<String>

Gets applications with an associated user identified by their user name (in the form 'domain\user'). If the 'UserFilterEnabled' property is true then access to the application is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserUPN**<String>

Gets applications with an associated user identified by their user principle name (in the form 'user@domain'). If the 'UserFilterEnabled' property is true then access to the application is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrowserName**<String>

Gets only the applications that match the supplied name. The BrowserName is usually an internal name for the application and is unique in the site.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Client Folder**<String>

Gets only the applications that match the specified value for the folder the application belongs to as seen by the end-user.

This folder can be seen in the Citrix Online Plug-in, in Web Services, and also potentially in the user's start menu.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CommandLineArguments**<String>

Gets only the applications that match the supplied arguments to the command-line executable.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CommandLineExecutable**<String>

Gets only the applications that match the supplied command-line executable.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CpuPriorityLevel**<CpuPriorityLevel>

Gets only the applications that have the specified value for the CPU priority level of the launched executable. Valid values are: Low, BelowNormal, Normal, AboveNormal, and High.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Gets only the applications that match the supplied description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Gets only the applications that have the specified value for whether the application is enabled. Disabled applications are still visible to users (that is controlled by the Visible setting) but cannot be launched.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconFromClient**<Boolean>

Gets only the applications that have the specified value for whether the application icon should be retrieved from the user device.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconUid**<Int32>

Gets only the applications that use the specified icon (identified by its Uid).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MetadataKey**<String>

Gets only applications whose associated metadata contains key names matching the specified value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName**<String>

Gets applications whose published name matches the supplied pattern.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureCmdLineArgumentsEnabled**<Boolean>

Gets only the applications that have the specified value for whether the command-line arguments should be secured. This is reserved for possible future use, and all applications of type HostedOnDesktop can only have this value set to true.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ShortcutAddedToDesktop**<Boolean>

Gets only the applications that match depending on whether a shortcut for the application has been added to the user device or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ShortcutAddedToStart Menu**<Boolean>

Gets only the applications that match depending on whether a shortcut for the application has been added to Start Menu of the user device or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Start MenuFolder**<String>

Gets only the applications that match the specified name for the start menu folder that holds the application shortcut. This is valid only for the Citrix Online Plug-in.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserFilterEnabled**<Boolean>

Gets only applications whose user filter is in the specified state.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UUID**<Guid>

Gets applications with the specified value of UUID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Visible**<Boolean>

Gets only the applications that have the specified value for whether it is visible to the users.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WaitForPrinterCreation**<Boolean>

Gets only the applications that match depending on whether the VDA delays starting the application until printers are set up.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WorkingDirectory**<String>

Gets only the applications that match the specified working directory.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-DesktopUid<Int32>**

Gets only the applications that have been associated (using a desktop group) to the specified desktop (identified by its Uid). Note that an application is not directly associated with a desktop, but only indirectly by which desktop group it has been published to.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUid<Int64>**

Gets only the applications that are running in the specified session (identified by its Uid).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserSID<String>**

Gets only applications with their accessibility restricted to include the specified user.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid<Int32>**

Gets only the applications that have been published to the specified desktop group (identified by its Uid).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|               |                                                         |
|---------------|---------------------------------------------------------|
| Required?     | false                                                   |
| Default Value | The default sort order is by name or unique identifier. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.Application

Get-BrokerApplication returns an object for each application it gets.

## Notes

Get-BrokerApplication returns just the application object, and as such is not a complete picture. The returned objects do not tell you what File-Type Associations are configured for this application, etc.

Use the following cmdlets to gather data related to applications (shown with examples of syntax):

```
Get-BrokerConfiguredFTA -ApplicationUid $app.Uid
```

```
Get-BrokerTag -ApplicationUid $app.Uid
```

```
Get-BrokerDesktopGroup -ApplicationUid $app.Uid
```

```
Get-BrokerDesktop -PublishedApplication $app
```

```
Get-BrokerSession -ApplicationUid $app.Uid
```

```
Get-BrokerApplicationInstance -ApplicationUid $app.Uid
```

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerApplication Notepad
```

Returns the application with the Name of "Notepad".

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerApplication -PublishedName Note* -Enabled $true
```

Returns the applications that have a PublishedName starting with "Note" and that are enabled.

# Get-BrokerApplicationInstance

Jan 04, 2017

Gets the running applications on the desktops.

## Syntax

```
Get-BrokerApplicationInstance -Uid <Int64> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerApplicationInstance [-ApplicationName <String>] [-ApplicationUid <Int32>] [-ApplicationUUID <Guid>] [-Instances <Int32>] [-MachineName <String>] [-MachineUid <Int32>] [-Metadata <String>] [-SessionKey <Guid>] [-SessionUid <Int64>] [-UserName <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Get-BrokerApplicationInstance gets the published applications that are running on desktops.

Only published applications that are launched from a Citrix client are returned. If a user launches an application from within a session (by double-clicking on an attachment from an email, for example) this will not show up in the list of running applications.

Also note that this is a list of launched published applications, not a list of processes running on the desktop. In some cases the original process associated with the published application may no longer be running, but if the session is still running the published application may be listed as running.

The number of instances for each published application running in a session is also returned. For example, if a user launches two Notepad applications from a Citrix client, and session-sharing occurs such that both Notepad applications run in the same session, then the Instances property indicates that 2 copies are running in the session.

See Get-BrokerApplication and Get-BrokerSession to get the details for the applications and sessions, respectively.

The Get-BrokerMachine cmdlet also returns a list of published applications that are running on a desktop. See the "ApplicationsInUse" attribute of the returned desktop objects.

----- BrokerApplicationInstance Object

The BrokerApplicationInstance object represents an instance of a published application in the site. It contains the following properties:

-- ApplicationName (System.String)

The administrative name of the application.

-- ApplicationUid (System.Int32)

The UID of the application.

-- ApplicationUUID (System.Guid)

The UUID of the application.

-- Instances (System.Int32)

The number of times this published application is running in the specified session.

-- MachineName (System.String)

Machine's SAM name (of the form domain\machine). If SAM name is unavailable, contains the machine's SID.

-- MachineUid (System.Int32)

UID of underlying machine.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Metadata for this application instance.

-- SessionKey (System.Guid)

The key of the session.

-- SessionUid (System.Int64)

The UID of the session.

-- Uid (System.Int64)

The unique identifier for this application instance object itself, distinct from the Uids of either application or session objects.

-- UserName (System.String)

User name (SAMName).

## Related topics

[Get-BrokerApplication](#)

[Get-BrokerDesktop](#)

[Get-BrokerSession](#)

## Parameters

**-Uid**<Int64>

Gets only the application instances specified by the unique identifier. This is the unique identifier for the application instance object itself, and is distinct from the Uids of either application or session objects.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

---

**-ApplicationName**<String>

Gets only application instances for the specified application name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationUid**<Int32>

Gets only application instances for the specified application Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationUUID**<Guid>

Gets only the application instances for the specified application UUID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Instances**<Int32>

Gets only application instances that match the specified number of instances.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName**<String>

Gets only application instances running on the specified machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineUid**<Int32>

Gets only application instances running on the machine with the specified UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionKey**<Guid>

Gets only application instances for the published applications running in the specified session.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-SessionUid**<Int64>

Gets only application instances for the published applications running in the specified session.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserName**<String>

Gets only application instances being run by the specified users.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.ApplicationInstance

Get-BrokerApplicationInstance returns an object for each application instance it gets.

Examples

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerApplicationInstance -Uid 3

Returns the application instance with a Uid of 3. Note that this is the unique identifier for application instances, which is distinct from the unique identifiers of either application or session objects.

----- **EXAMPLE 2** -----

C:\PS> \$app = Get-BrokerApplication Notepad

C:\PS> Get-BrokerApplicationInstance -ApplicationUid \$app.Uid

Returns all the application instances for the Notepad application. Use this to see if there are any launched instances of Notepad running in your site and, if so, from which desktops.

----- **EXAMPLE 3** -----

C:\PS> \$sessions = Get-BrokerSession -MachineName "ACME\Worker1"

```
C:\PS> for ($i=0; $i -lt $sessions.Length; $i++) {
    Get-BrokerApplicationInstance -SessionUid $sessions[$i].SessionUid
}
```

Returns all the applications that are running on the "Worker1" machine in the "ACME" domain. Use this to see which published applications are running on a specific machine.

Note that the SessionUid, not the SessionId, is specified as a parameter to this cmdlet. The SessionId is a unique identifier that Remote Desktop Services uses to track the session, and is unique only on that machine. The SessionUid, on the other hand, is unique across the entire site.

The "ApplicationsInUse" attribute of the returned session object also provides a list of running launched applications, and in many cases might be more convenient to use. It returns a list of application BrowserNames.

# Get-BrokerAssignmentPolicyRule

Jan 04, 2017

Gets desktop rules from the site's assignment policy.

## Syntax

```
Get-BrokerAssignmentPolicyRule [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerAssignmentPolicyRule [[-Name] <String>] [-ColorDepth <ColorDepth>] [-Description <String>] [-DesktopGroupUid <Int32>] [-Enabled <Boolean>] [-ExcludedUser <User>] [-ExcludedUserFilterEnabled <Boolean>] [-IconUid <Int32>] [-IncludedUser <User>] [-IncludedUserFilterEnabled <Boolean>] [-MaxDesktops <Int32>] [-Metadata <String>] [-PublishedName <String>] [-SecureIcaRequired <Boolean>] [-UUID <Guid>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns desktop rules matching the specified search criteria from the site's assignment policy. If no search criteria are specified, all desktop rules in the assignment policy are obtained.

A desktop rule in the assignment policy defines the users who are entitled to self-service persistent machine assignments from the rule's desktop group. A rule defines how many machines a user is allowed from the group for delivery of full desktop sessions.

----- BrokerAssignmentPolicyRule Object

The `BrokerAssignmentPolicyRule` object represents a single desktop rule within the site's assignment policy. It contains the following properties:

-- `ColorDepth` (Citrix.Broker.Admin.SDK.ColorDepth?)

The color depth of desktop sessions launched by the user from machines assigned to them by the rule. If null, the equivalent setting from the rule's desktop group is used.

-- `Description` (System.String)

Optional description of the rule. The text may be visible to the end user, for example, as a tooltip associated with the desktop entitlement.

-- `DesktopGroupUid` (System.Int32)

The unique ID of the desktop group to which the rule applies.

-- `Enabled` (System.Boolean)

Indicates whether the rule is enabled. A disabled rule is ignored when evaluating the site's assignment policy.

-- `ExcludedUserFilterEnabled` (System.Boolean)

Indicates whether the excluded users filter is enabled. If the filter is disabled then any user entries in the filter are ignored when assignment policy rules are evaluated.

-- ExcludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

The excluded users filter of the rule, that is, the users and groups who are explicitly denied entitlements to machine assignments from the rule's desktop group.

-- IconUid (System.Int32?)

The unique ID of the icon used for the machine entitlement seen by the user or, after a machine is assigned by the rule, the icon for the desktop itself. If null, the equivalent setting from the rule's desktop group is used.

-- IncludedUserFilterEnabled (System.Boolean)

Indicates whether the included users filter is enabled. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly entitled to the machine assignments described by the rule.

For rules that relate to RemotePC desktop groups however, if the included user filter is disabled, the rule is effectively disabled.

-- IncludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

The included users filter of the rule, that is, the users and groups who are entitled to machine assignments from the rule's desktop group.

-- MaxDesktops (System.Int32)

The number of machines from the rule's desktop group to which a user is entitled. Where an entitlement is granted to a user group rather than an individual, the number of machines applies to each member of the user group independently.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

A collection of arbitrary key/value pairs that can be associated with the rule. The administrator can use these values for any purpose; they are not used by the site itself in any way.

-- Name (System.String)

The administrative name of the rule. Each rule in the site's assignment policy must have a unique name (irrespective of whether they are desktop or application rules).

-- PublishedName (System.String)

The published name of the desktop entitlement seen by the user or, after a machine is assigned by the rule, the published name of the desktop itself. If null, the equivalent setting from the rule's desktop group is used.

-- SecureIcaRequired (System.Boolean?)

Indicates whether the rule requires the SecureICA protocol for desktop sessions launched using a machine assigned by the rule. If null, the equivalent setting from the rule's desktop group is used.

-- Uid (System.Int32)

The unique ID of the rule itself.

-- UUID (System.Guid)

UUID of the rule.

## Related topics

[New-BrokerAssignmentPolicyRule](#)

[Set-BrokerAssignmentPolicyRule](#)

[Rename-BrokerAssignmentPolicyRule](#)

[Remove-BrokerAssignmentPolicyRule](#)

## Parameters

**-Uid**<Int32>

Gets the desktop rule with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Name**<String>

Gets only desktop rules with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ColorDepth**<ColorDepth>

Gets only desktop rules with the specified color depth.

Valid values are \$null, FourBit, EightBit, SixteenBit, and TwentyFourBit.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description<String>**

Gets only desktop rules with the specified description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid<Int32>**

Gets only desktop rules that apply to the desktop group with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled<Boolean>**

Gets only rules that are in the specified state, either enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUser<User>**

Gets only desktop rules that have the specified user in their excluded users filter (whether the filter is enabled or not).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUserFilterEnabled<Boolean>**



Gets only desktop rules that have their excluded user filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconUid**<Int32>

Gets only desktop rules using the icon with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUser**<User>

Gets only desktop rules that have the specified user in their included users filter (whether the filter is enabled or not).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUserFilterEnabled**<Boolean>

Gets only desktop rules that have their included user filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxDesktops**<Int32>

Gets only desktop rules granting the specified number of machine assignment entitlements.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName**<String>

Gets only desktop rules with the specified published name, that is, the desktop name that the end user sees.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaRequired**<Boolean>

Gets only desktop rules that require desktop sessions to machines assigned by the rule to use the SecureICA protocol (\$true) or not (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UUID**<Guid>

Gets rules with the specified value of UUID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See `about_Broker_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|               |                                                                                        |
|---------------|----------------------------------------------------------------------------------------|
| Required?     | false                                                                                  |
| Default Value | localhost. Once a value is provided by any cmdlet, this value will become the default. |
|               |                                                                                        |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

Citrix.Broker.Admin.SDK.AssignmentPolicyRule

Get-BrokerAssignmentPolicyRule returns all assignment policy rules that match the specified selection criteria.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerAssignmentPolicyRule
```

Returns all desktop rules from the assignment policy. This offers a complete description of the current site's assignment policy with respect to machine assignment entitlements for delivery of desktop sessions from private desktop groups.

### ----- EXAMPLE 2 -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Sales Support'
```

```
C:\PS> Get-BrokerAssignmentPolicyRule -DesktopGroupUid $dg.Uid
```

Returns all rules in the assignment policy that give users entitlements to machine assignments in the Sales Support desktop group for delivery of full desktop sessions.

# Get-BrokerCatalog

Jan 04, 2017

Gets catalogs configured for this site.

## Syntax

```
Get-BrokerCatalog [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerCatalog [[-Name] <String>] [-AllocationType <AllocationType>] [-AssignedCount <Int32>] [-AvailableAssignedCount <Int32>] [-AvailableCount <Int32>] [-AvailableUnassignedCount <Int32>] [-Description <String>] [-HypervisorConnectionUid <Int32>] [-IsRemotePC <Boolean>] [-MachinesArePhysical <Boolean>] [-Metadata <String>] [-MinimumFunctionalLevel <FunctionalLevel>] [-PersistUserChanges <PersistUserChanges>] [-ProvisioningSchemeId <Guid>] [-ProvisioningType <ProvisioningType>] [-PvsAddress <String>] [-PvsDomain <String>] [-RemotePCDesktopGroupPriority <Int32>] [-RemotePCDesktopGroupUid <Int32>] [-RemotePCHypervisorConnectionUid <Int32>] [-ScopeId <Guid>] [-ScopeName <String>] [-SessionSupport <SessionSupport>] [-UnassignedCount <Int32>] [-UsedCount <Int32>] [-UUID <Guid>] [-MachineUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves catalogs matching the specified criteria. If no parameters are specified this cmdlet enumerates all catalogs.

See `about_Broker_Filtering` for information about advanced filtering options.

----- BrokerCatalog Object

The catalog object returned represents a group of related physical or virtual machines that have been configured in the site.

See `about_Broker_Machines` for more information.

-- AllocationType (Citrix.Broker.Admin.SDK.AllocationType)

Denotes how the machines in the catalog are allocated to a user.

Possible values are:

- o Static: Machines get assigned to a user either by the admin or on first use. This relationship is static and changes only if an admin explicitly changes the assignments.
- o Random: Machines are allocated to users randomly from a pool of available machines.

-- AssignedCount (System.Int32)

The number of assigned machines (machines that have been assigned to a user/users or a client name/address).

-- AvailableAssignedCount (System.Int32)

The number of available machines (not in a desktop group), that are also assigned to users.

-- AvailableCount (System.Int32)

The number of available machines (those not in any desktop group).

-- AvailableUnassignedCount (System.Int32)

The number of available machines (those not in any desktop group) that are not assigned to users.

-- Description (System.String)

Description of the catalog.

-- HypervisorConnectionUid (System.Int32?)

The Uid of the hypervisor connection that is associated with the machines in the catalog. This property only applies to MCS provisioned catalogs. For other provisioning types machines can be from one or more different hypervisor connections.

-- IsRemotePC (System.Boolean)

Specifies whether or not the catalog is a RemotePC catalog. Remote PC catalogs automatically configure appropriate machines without the need for manual configuration. See about\_Broker\_RemotePC for more information.

-- MachinesArePhysical (System.Boolean)

Specifies whether or not the machines in the catalog can be power-managed by the broker.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Holds any metadata associated with the catalog.

-- MinimumFunctionalLevel (Citrix.Broker.Admin.SDK.FunctionalLevel)

The expected minimal functional level of the machines in the catalog.

-- Name (System.String)

Name of the catalog.

-- PersistUserChanges (Citrix.Broker.Admin.SDK.PersistUserChanges)

Specifies how user changes are persisted on machines in the catalog. Possible values are:

o OnLocal: User changes are stored on the machine's local storage.

o Discard: User changes are discarded.

o OnPvd: User changes are stored on the user's personal vDisk.

-- ProvisioningSchemeId (System.Guid?)

The GUID of the provisioning scheme (if any) associated with the catalog. This only applies if the provisioning type is MCS.

-- ProvisioningType (Citrix.Broker.Admin.SDK.ProvisioningType)

Specifies how the machines are provisioned in the catalog. Possible values are:

o Manual: No provisioning.

o PVS: Machine provisioned by PVS (machine may be physical, blade, VM...)

o MCS: Machine provisioned by MCS (machine must be a VM).

-- PvsAddress (System.String)

IP address of the PVS server to be used in a catalog with a PVS ProvisioningType.

-- PvsDomain (System.String)

The domain of the PVS server to be used in a catalog with a PVS ProvisioningType.

-- RemotePCDesktopGroupPriorities (System.Int32[])

Remote PC desktop groups' association priorities.

-- RemotePCDesktopGroupUids (System.Int32[])

UIDs of the Remote PC desktop groups associated with this catalog.

-- RemotePCHypervisorConnectionUid (System.Int32?)

UID of the hypervisor connection used for powering on RemotePC machines in this catalog (only for catalogs with IsRemotePC set to true).

-- Scopes (Citrix.Broker.Admin.SDK.ScopeReference[])

The list of the delegated admin scopes to which the catalog belongs.

-- SessionSupport (Citrix.Broker.Admin.SDK.SessionSupport)

Specifies the session support of the machines in the catalog. Valid values are:

SingleSession, MultiSession.

-- Uid (System.Int32)

Uid of the catalog.

-- UnassignedCount (System.Int32)

The number of unassigned machines (machines not assigned to users).

-- UsedCount (System.Int32)

The number of machines in the catalog that are in a desktop group.

-- UUID (System.Guid)

The global ID of the catalog.

Related topics

[New-BrokerCatalog](#)

[Remove-BrokerCatalog](#)

[Rename-BrokerCatalog](#)



## Set-BrokerCatalog

### Parameters

#### **-Uid**<Int32>

Get catalogs with the specified UID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-Name**<String>

Gets catalogs with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-AllocationType**<AllocationType>

Gets catalogs that are of the specified allocation type. Values can be:

- o Static - Machines in a catalog of this type are permanently assigned to a user.
- o Permanent - equivalent to 'Static'.
- o Random - Machines in a catalog of this type are picked at random and temporarily assigned to a user.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-AssignedCount**<Int32>

Gets catalogs containing a specified number of assigned machines (machines that have been assigned to users).

This property is typically used with advanced filtering; see `about_Broker_Filtering`.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AvailableAssignedCount**<Int32>

Gets catalogs containing a specified number of available machines (those not in any desktop group) that are also assigned to users.

This property is typically used with advanced filtering; see about\_Broker\_Filtering.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AvailableCount**<Int32>

Gets catalogs containing a specified number of available machines (those not in any desktop group).

This property is typically used with advanced filtering; see about\_Broker\_Filtering.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AvailableUnassignedCount**<Int32>

Gets catalogs containing a specified number of available machines (those not in any desktop group) that are not assigned to users.

This property is typically used with advanced filtering; see about\_Broker\_Filtering.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

|  |  |
|--|--|
|  |  |
|--|--|

**-Description**<String>

Gets catalogs with the specified description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionUid**<Int32>

Gets catalogs associated with the specified hypervisor connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsRemotePC**<Boolean>

Gets catalogs with the specified IsRemotePC value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachinesArePhysical**<Boolean>

Specifies whether machines in the catalog can be power-managed by the Citrix Broker Service. Where the Citrix Broker Service cannot control the power state of the machine specify \$true, otherwise \$false. Can only be specified together with a provisioning type of Pvs or Manual, or if used with the legacy CatalogKind parameter only with Pvs or PvsPvd catalog kinds.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MinimumFunctionalLevel**<FunctionalLevel>

Gets catalogs with a specific MinimumFunctionalLevel.

Valid values are L5, L7, L7\_6

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PersistUserChanges**<PersistUserChanges>

Gets catalogs with the specified behavior when persisting changes made by the end user. Possible values are:

- o OnLocal - User changes are stored on the machine's local storage.
- o Discard - User changes are discarded.
- o OnPvd - User changes are stored on the user's personal vDisk.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ProvisioningSchemeld**<Guid>

Gets catalogs associated with the specified provisioning scheme.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ProvisioningType**<ProvisioningType>

Specifies the provisioning type for the catalog. Values can be:

- o Manual - No provisioning.
- o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).
- o MCS - Machine provisioned by MCS (machine must be VM).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvsAddress**<String>

Gets catalogs containing machines provided by the Provisioning Services server with the specified address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvsDomain**<String>

Gets catalogs containing machines provided by the Provisioning Services server in the specified domain.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemotePCDesktopGroupPriority<Int32>**

Gets Remote PC catalogs with a Remote PC desktop group association with the specified priority.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemotePCDesktopGroupUid<Int32>**

Gets Remote PC catalogs associated with the specified Remote PC desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemotePCHypervisorConnectionUid<Int32>**

Gets Remote PC catalogs associated with the specified Remote PC hypervisor connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ScopeId<Guid>**

Gets catalogs that are associated with the given scope identifier.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ScopeName**<String>

Gets catalogs that are associated with the given scope name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionSupport**<SessionSupport>

Gets catalogs that have the specified session capability. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UnassignedCount**<Int32>

Gets catalogs containing a specified number of unassigned machines (machines not assigned to users).

This property is typically used with advanced filtering; see about\_Broker\_Filtering.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UsedCount**<Int32>

Gets catalogs containing a specified number of machines used in a desktop group.

This property is typically used with advanced filtering; see about\_Broker\_Filtering.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? |       |

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UUID**<Guid>

Get catalogs with the specified global ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineUid**<Int32>

Gets the catalog containing the machine referenced by the specified unique identifier (UID).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |       |
|------------------------|-------|
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None

Return Values

Citrix.Broker.Admin.SDK.Catalog

Get-BrokerCatalog returns an object for each matching catalog.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerCatalog -AllocationType Random
```

```
C:\PS> Get-BrokerCatalog -Filter { AllocationType -eq 'Random'}
```

These commands return all catalogs containing machines that are randomly assigned to users. The second form uses advanced filtering (see about\_Broker\_Filtering).

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerCatalog -Filter { AvailableCount -gt 10}
```

This command returns catalogs with more than 10 unused machines that are available for assignment to users.

----- **EXAMPLE 3** -----

```
C:\PS> Get-BrokerCatalog -MaxRecordCount 1 -ProvisioningType Manual -SortBy '-AvailableCount'
```

This command returns the unmanaged catalog with the highest number of available machines.

# Get-BrokerConfigurationSlot

Jan 04, 2017

Gets configuration slots configured for this site.

## Syntax

```
Get-BrokerConfigurationSlot [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerConfigurationSlot [[-Name] <String>] [-Metadata <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Get the list of configuration slots defined for this site. Each configuration slot determines a collection of related settings that can be specified in a machine configuration associated with this slot.

For example, a configuration slot may be defined to configure only "User Profile Manager" settings.

See [about\\_Broker\\_Filtering](#) for information about advanced filtering options.

----- BrokerConfigurationSlot Object

The configuration slot object returned represents a named collection of related settings.

-- Description (System.String)

Optional description of this configuration slot.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

A map of metadata associated with this configuration slot.

-- Name (System.String)

Unique name of this configuration slot.

-- SettingsGroup (System.String)

The encoded identity of the settings group that every setting in the associated machine configuration instances must belong to.

-- Uid (System.Int32)

Unique Uid of this configuration slot.

## Related topics

[New-BrokerConfigurationSlot](#)

[Remove-BrokerConfigurationSlot](#)

[Get-BrokerMachineConfiguration](#)

## Parameters

### **-Uid**<Int32>

Get only the configuration slot with the specified unique identifier.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-Name**<String>

Get only the configuration slot with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: `-Metadata "abc:x*"` matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See `about_Broker_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? |       |

|                        |       |
|------------------------|-------|
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.ConfigurationSlot

Get-BrokerConfigurationSlot returns an object for each matching slot.

**Examples**

----- **EXAMPLE 1** -----

Get-ConfigurationSlot  
Retrieves every configuration slot.

----- **EXAMPLE 2** -----

Get-ConfigurationSlot -Name "AppV"

Retrieves the configuration slot named "AppV".

# Get-BrokerConfiguredFTA

Jan 04, 2017

Gets any file type associations configured for an application.

## Syntax

```
Get-BrokerConfiguredFTA -Uid <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerConfiguredFTA [-ApplicationUid <Int32>] [-ContentType <String>] [-ExtensionName <String>] [-HandlerDescription <String>] [-HandlerName <String>] [-HandlerOpenArguments <String>] [-UUID <Guid>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets any file type associations that are configured for content redirection to a published application.

File type association associates a file extension (such as ".txt") with an application (such as Notepad). In a Citrix environment file type associations on a user device can be configured so that when an user clicks on a document it launches the appropriate published application. This is known as "content redirection".

Configured file type associations are different from imported file type associations. Configured file type associations are those that are actually associated with published applications for the purposes of content redirection. Imported file type associations are lists of known file type associations for a given desktop group. See Update-BrokerImportedFTA for more information about imported file type associations.

### ----- BrokerConfiguredFTA Object

The BrokerConfiguredFTA object represents a file type association configured for a published application. It contains the following properties:

-- ApplicationUid (System.Int32)

The Uid of the application configured for the file type association.

-- ContentType (System.String)

Content type of the file, such as "text/plain" or "application/vnd.ms-excel".

-- ExtensionName (System.String)

A single file extension, such as .txt, unique within the scope of a desktop group.

-- HandlerDescription (System.String)

File type description, such as "Test Document", "Microsoft Word Text Document", etc.

-- HandlerName (System.String)

File type handler name, e.g. "Word.Document.8" or TXTFILE.

-- HandlerOpenArguments (System.String)



The arguments used for the 'open' action on files of this type.

-- Uid (System.Int32)

Unique internal identifier of configured file type association.

-- UUID (System.Guid)

UUID of the configured file type association.

## Related topics

[New-BrokerConfiguredFTA](#)

[Remove-BrokerConfiguredFTA](#)

[Update-BrokerImportedFTA](#)

## Parameters

**-Uid**<Int32>

Gets only the configured file type association for the specified unique identifier.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationUid**<Int32>

Gets only the configured file type associations for the specified application unique identifier.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ContentType**<String>

Gets only the configured file type associations for the specified content type (as seen in the Registry). For example, "text/plain" or "application/msword".

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExtensionName**<String>

Gets only the configured file type associations for the specified extension name. For example, ".txt" or ".doc".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HandlerDescription**<String>

Gets only the configured file type associations for the specified handler description. For example, "Text Document".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HandlerName**<String>

Gets only the configured file type associations for the specified handler name. For example, "TXTFILE" or "Word.Document.8".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HandlerOpenArguments**<String>

Gets only the configured file type associations for the specified open argument to the handler. For example, "%1".

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-UUID**<Guid>

Gets configured file type associations with the specified value of UUID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

None No input is accepted from the pipeline.

### Return Values

Citrix.Broker.Admin.SDK.ConfiguredFTA

This cmdlet returns one or more ConfiguredFTA objects.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerConfiguredFTA
Returns all configured file type associations.
```

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerConfiguredFTA -ExtensionName ".txt"
Returns only configured file type associations that have a ".txt" extension.
```

# Get-BrokerConnectionLog

Jan 04, 2017

Get entries from the site's session connection log.

## Syntax

```
Get-BrokerConnectionLog [-Uid] <Int64> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerConnectionLog [[-MachineName] <String>] [-BrokeringTime <DateTime>] [-BrokeringUserName <String>] [-BrokeringUserUPN <String>] [-ConnectionFailureReason <ConnectionFailureReason>] [-Disconnected <Boolean>] [-EndTime <DateTime>] [-EstablishmentTime <DateTime>] [-MachineDNSName <String>] [-MachineUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets connection log entries matching the specified criteria. If no parameters are specified all connection log entries are returned.

The session connection log contains entries describing each brokered connection, or reconnection, attempt to a session in the site.

Each log entry describes a single connection brokering attempt to a new or existing session within the site. A single session can have multiple entries in the connection log, for example where the end user brokers a connection to a new session, disconnects and later brokers a reconnection. Conversely, other sessions may have none (e.g. console sessions).

By default connection log entries are removed after 48 hours.

For information about advanced filtering options when using the `-Filter` parameter, see `about_Broker_Filtering`; for information about machines, see `about_Broker_Machines`.

### ----- BrokerConnectionLog Object

The `BrokerConnectionLog` object represents a single brokered connection attempt to a new or existing session on a machine in the site. It contains the following properties:

-- `BrokeringTime` (System.DateTime)

The time at which the connection attempt was made.

-- `BrokeringUserName` (System.String)

The name of the user making the connection (in DOMAIN\User format).

-- `BrokeringUserUPN` (System.String)

The name of the user making the connection (in user@upndomain.com format).

-- `ConnectionFailureReason` (Citrix.Broker.Admin.SDK.ConnectionFailureReason?)

The status of the connection attempt. A value of `None` indicates that the connection was successfully established, `$null`

that the attempt is still in progress, and other values indicate that the attempt failed for the specified reason.

-- Disconnected (System.Boolean?)

Indicates if the connection was ended by disconnection (True), logoff or establishment failure (False), or is still active (\$null).

-- EndTime (System.DateTime?)

The time at which the connection ended. If the connection ended by disconnection, the underlying machine session would still exist in a disconnected state.

-- EstablishmentTime (System.DateTime?)

The time at which the connection was successfully established. The value is \$null if the connection attempt failed or is still in progress.

-- MachineDNSName (System.String)

The name of the machine to which the connection was made (in machine@dnsdomain.com form).

-- MachineName (System.String)

The name of the machine to which the connection was made (in DOMAIN\Machine format).

-- MachineUid (System.Int32)

The UID of the machine to which the connection was made.

-- Uid (System.Int64)

The UID of the connection log entry itself.

## Related topics

### Parameters

**-Uid**<Int64>

Gets a specific connection log entry identified by its UID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName**<String>

Gets connection log entries for the specified machines (in DOMAIN\Machine format).

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrokeringTime**<DateTime>

Gets connection log entries with a specified brokering time. For more flexibility when searching on brokering time use the -Filter parameter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrokeringUserName**<String>

Gets connection log entries for the specified users (in DOMAIN\User format).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrokeringUserUPN**<String>

Gets connection log entries for the specified users (in user@upndomain.com format).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ConnectionFailureReason**<ConnectionFailureReason>

Gets connection log entries which failed for the specified reason.

|           |       |
|-----------|-------|
| Required? | false |
|           |       |



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Disconnected**<Boolean>

Gets connection log entries with the specified disconnection status, that is, whether the connection was disconnected, or logged-off.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-EndTime**<DateTime>

Gets connection log entries with the specified end time. For more flexibility when searching on end time use the -Filter parameter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-EstablishmentTime**<DateTime>

Gets connection log entries with the specific establishment time. For more flexibility when searching on establishment time use the -Filter parameter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineDNSName**<String>

Gets connection log entries for the specified machines (in machine@dnsdomain.com format).

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineUid**<Int32>

Gets connection log entries for a specific machine identified by its UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See `about_Broker_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host

name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

None You cannot pipe input into this cmdlet.

### Return Values

Citrix.Broker.Admin.SDK.ConnectionLog

An entry from the connection log.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> $when = [DateTime]::Now - [TimeSpan]::FromMinutes(30)
```

```
C:\PS> Get-BrokerConnectionLog -Filter {BrokeringTime -gt $when} -SortBy '+MachineName,-EndTime'
```

Gets all connection log entries for sessions brokered in the past 30 minutes, ordered first by machine name (ascending), then by session end time (descending).

# Get-BrokerController

Jan 04, 2017

Gets Controllers running broker services in the site.

## Syntax

```
Get-BrokerController [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerController [-MachineName] <String> [-ControllerVersion <String>] [-DesktopsRegistered <Int32>] [-DNSName <String>] [-LastActivityTime <DateTime>] [-LastLicensingServerEvent <LicensingServerEvent>] [-LastLicensingServerEventTime <DateTime>] [-LastStartTime <DateTime>] [-LicensingGraceState <LicensingGraceState>] [-LicensingServerState <LicensingServerState>] [-Metadata <String>] [-OSType <String>] [-OSVersion <String>] [-SID <String>] [-State <ControllerState>] [-UUID <Guid>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets Controllers from the current site that match the specified search criteria.

Controllers are server machines running instances of the broker service. The broker service is responsible for the brokering of user sessions to desktops or applications, and for power management of the underlying machines. An operational site must contain at least one Controller.

If no search criteria are specified, all Controllers in the site are obtained.

----- BrokerController Object

The BrokerController object represents a single controller running an instance of the Broker service. It contains the following properties:

-- ActiveSiteServices (System.String[])

The Broker site services active on the controller.

-- AssociatedHypervisorConnectionUids (System.Int32[])

The UIDs of the hypervisor connections being managed by the Broker service on the controller.

-- ControllerVersion (System.String)

The version of the Broker service on the controller.

-- DesktopsRegistered (System.Int32)

The number of VDA machines registered with the Broker service on the controller.

-- DNSName (System.String)

The DNS name of the controller.

-- LastActivityTime (System.DateTime?)

The last reported activity time of the Broker service on the controller.

-- LastLicensingServerEvent (Citrix.Broker.Admin.SDK.LicensingServerEvent?)

Last significant licensing server event reported by the Broker service on the controller.

-- LastLicensingServerEventDetails (System.String[])

Additional details associated with the last significant licensing server event.

-- LastLicensingServerEventTime (System.DateTime?)

Time at which the last significant licensing server event was reported.

-- LastStartTime (System.DateTime?)

The last start-up time of the Broker service on the controller.

-- LicensingGracePeriodReasons (Citrix.Broker.Admin.SDK.LicensingGracePeriodReason[])

Current active or expired licensing grace periods in effect on the controller.

-- LicensingGracePeriodTimesRemaining (System.TimeSpan[])

Times remaining in currently active or expired licensing grace periods in effect on the controller. Expired grace periods are indicated by zero remaining time. The number and order of entries in this list matches that in the LicensingGracePeriodReasons list.

-- LicensingGraceState (Citrix.Broker.Admin.SDK.LicensingGraceState)

The licensing grace state currently in effect in the Broker service on the controller.

-- LicensingServerState (Citrix.Broker.Admin.SDK.LicensingServerState)

The licensing server state currently in effect in the Broker service on the controller.

-- MachineName (System.String)

The Windows name of the controller.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

The metadata for the controller.

-- OSType (System.String)

The Operating System type of the controller.

-- OSVersion (System.String)

The Operating System version of the controller.

-- SID (System.String)

The SID of the controller.

-- State (Citrix.Broker.Admin.SDK.ControllerState)

The state of the Broker service on the controller.

-- Uid (System.Int32)

The UID of the controller instance.

-- UUID (System.Guid)

A globally unique identifier of the controller instance.

## Related topics

[Get-BrokerDesktop](#)

## Parameters

**-Uid**<Int32>

Gets only Controller with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName**<String>

Gets only Controllers with the specified Windows name. ('domain\machine')

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ControllerVersion**<String>

Gets only Controllers running the specified version of the broker service.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopsRegistered<Int32>**

Gets only Controllers that have the specified number of desktops currently registered. This parameter is mainly of use with advanced filtering; see about\_Broker\_Filtering.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DNSName<String>**

Gets only Controllers with the specified DNS name ('machine.domain')

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastActivityTime<DateTime>**

Gets only Controllers last reported as active at the specified time. This parameter is mainly of use with advanced filtering; see about\_Broker\_Filtering.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastLicensingServerEvent<LicensingServerEvent>**

Gets only Controllers with the specified last license server event recorded.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-LastLicensingServerEventTime**<DateTime>

Gets only Controllers with its last recorded licensing server event at the specified time. This parameter is mainly of use with advanced filtering; see about\_Broker\_Filtering.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastStartTime**<DateTime>

Gets only Controllers that last started-up at the specified time. This parameter is mainly of use with advanced filtering; see about\_Broker\_Filtering.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LicensingGraceState**<LicensingGraceState>

Gets only Controllers in the specified licensing grace state.

Valid values are: NotActive, InOutOfBoxGracePeriod, InSupplementalGracePeriod, InEmergencyGracePeriod and GracePeriodExpired.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LicensingServerState**<LicensingServerState>

Gets only Controllers in the specified licensing server state. Valid values are: ServerNotSpecified, NotConnected, OK, LicenseNotInstalled, LicenseExpired, Incompatible and Failed.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSType**<String>

Gets only Controllers running the specified Operating System type.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSVersion**<String>

Gets only Controllers running the specified Operating System version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SID**<String>

Gets only Controllers with the specified SID.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-State**<ControllerState>

Gets only Controllers currently in the specified state.

Valid values are: Failed, Off, On, and Active.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UUID**<Guid>

Gets only the Controller with the specified GUID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.Controller

Returns Controllers matching all specified selection criteria.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerController -State Active

Gets all Controllers in the site that are currently active (powered on and fully operational).

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerController -Filter 'LastStartTime -gt "-30:00"'

Gets all Controllers in the site that started-up in the last 30 minutes.

# Get-BrokerDBConnection

Jan 04, 2017

Gets the database connection string for the specified data store used by the Broker Service.

## Syntax

```
Get-BrokerDBConnection [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets the database connection string from the currently selected Broker Service instance.

If the returned string is blank, no valid connection string has been specified. In this case the service is running, but is idle and awaiting specification of a valid connection string.

The current service instance is the one on the local machine, or the one most recently specified using the `-AdminAddress` parameter of a Broker SDK cmdlet.

## Related topics

[Set-BrokerDBConnection](#)

[Get-BrokerServiceStatus](#)

[Test-BrokerDBConnection](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

System.String

The database connection string configured for the current Broker Service instance.

## Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### NoDBConnections

The database connection string for the Broker Service has not been specified.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerDBConnection -AdminAddress controller1.mydomain.net
```

Gets the database connection string in use by the Broker Service instance running on controller "controller1.mydomain.net".

# Get-BrokerDBSchema

Jan 04, 2017

Gets SQL scripts to create or maintain the database schema for the Citrix Broker Service.

## Syntax

```
Get-BrokerDBSchema -DatabaseName <String> [-ServiceGroupName <String>] [-ScriptType <DatabaseScriptType>] [-SID <String>] [-LocalDatabase] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets SQL scripts that can be used to create a new Citrix Broker Service database schema, add a new Broker service to an existing site, remove a Broker service from a site, or create a database server logon for a Broker service.

If no Sid parameter is provided, the scripts obtained relate to the currently selected Broker service instance, otherwise the scripts relate to Broker service instance running on the machine identified by the Sid provided. When obtaining the Evict script, a Sid parameter must be supplied.

The current service instance is the one on the local machine, or the one most recently specified using the -AdminAddress parameter of a Broker SDK cmdlet.

The service instance used to obtain the scripts does not need to be a member of a site or to have had its database connection configured.

The database scripts support only Microsoft SQL Server, or SQL Server Express, and require Windows integrated authentication to be used. They can be run using SQL Server's SQLCMD utility, or by copying the script into an SQL Server Management Studio (SSMS) query window and executing the query. If using SSMS, the query must be executed in 'SQLCMD mode'.

The ScriptType parameter determines which script is obtained. If ScriptType is not specified, or is FullDatabase, the script contains:

- o Creation of service schema
- o Creation of database server logon
- o Creation of database user
- o Addition of database user to Broker service roles

If ScriptType is Instance, the returned script contains:

- o Creation of database server logon
- o Creation of database user
- o Addition of database user to Broker service roles

If ScriptType is Evict, the returned script contains:

- o Removal of Broker service instance from database



o Removal of database user

If ScriptType is Login, the returned script contains:

o Creation of database server logon only

If the service uses two data stores they can exist in the same database.

You do not need to configure a database before using this command.

## Related topics

[Set-BrokerDBConnection](#)

[Test-BrokerDBConnection](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database into which the new Broker service schema is to be placed, or in which it already exists. The database itself is not created by any of the script types; it must already exist before the scripts are run.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ServiceGroupName**<String>

Specifies the name of the service group to be used when creating the database schema. The service group is a collection of all the Broker services that share the same database instance and are considered equivalent; that is, all the services within a service group can be used interchangeably.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ScriptType**<DatabaseScriptType>

Specifies the type of database script returned. Available script types are:

-- FullDatabase

Creates a database schema for the Citrix Broker Service in a database instance that does not already contain one. This is

used when creating a new site. DatabaseName and ServiceGroupName are required parameters for this script type.

-- Instance

Adds a Broker Service instance to a database and so to the associated site. Appropriate database server logons and users are created to allow the service instance access to the required service schemas.

-- Evict

Removes a Broker Service instance from the database and so from the site. All reference to the service instance is removed from the database. DatabaseName and Sid are required parameters for this script type.

-- Login

Adds a logon for the Broker Service instance to a database server. This is specifically for use when configuring SQL Server mirroring where the mirror server must have appropriate logons created for all service instances in the site.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SID**<String>

Specifies the SID of the controller on which the Broker Service instance to remove from the database is running (only valid for a script type of Evict).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | None  |
| Accept Pipeline Input? | false |

**-LocalDatabase**<SwitchParameter>

Specifies whether the database script is to be used in a database instance run on the same controller as other services in the service group. Including this parameter ensures the script creates only the required permissions for local services to access the database schema for Broker services. If this parameter is specified inappropriately, the service instance will not be able to connect to the database.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

System.String

A string containing the required SQL script for applying to a database.

**Notes**

The scripts returned support Microsoft SQL Server Express Edition, Microsoft SQL Server Standard Edition, and Microsoft SQL Server Enterprise Edition databases only, and are generated on the assumption that integrated authentication will be used.

If the ScriptType parameter is not included or set to 'FullDatabase', the full database script is returned, which will:

- Create the database schema.

- Create the user and the role (providing the schema does not already exist).

- Create the logon (providing the schema does not already exist).

If the ScriptType parameter is set to 'Instance', the script will:

- Create the user and the role (providing the schema does not already exist).

- Create the logon (providing the schema does not already exist) and associate it with a user.

If the ScriptType parameter is set to 'Login', the script will:

- Create the logon (providing the schema does not already exist) and associate it with a pre-existing user of the same name.

If the LocalDatabase parameter is included, the NetworkService account will be added to the list of accounts permitted to access the database. This is required only if the database is run on a controller.

If the command fails, the following errors can be returned.

## Error Codes

-----

### GetSchemasFailed

The database schema could not be found.

### ActiveDirectoryAccountResolutionFailed

The specified Active Directory account or Group could not be found.

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

### PermissionDenied

You do not have permission to execute this command.

### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

### CommunicationError

There was a problem communicating with the remote service.

### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS>Get-BrokerDBSchema -DatabaseName MySiteDB -ServiceGroupName MyServiceGroup > C:\BrokerSchema.sql  
Gets a script to create the full database schema for the Citrix Broker Service and copies it to a file called  
"C:\BrokerSchema.sql"
```

This script can be used to create the service schema in a database with name "MySiteDB", which must already exist, and must not already contain a Broker service schema.

----- **EXAMPLE 2** -----

```
C:\PS>Get-BrokerDBSchema -DatabaseName MySiteDB -ScriptType Login > C:\BrokerLogins.sql
```

Gets a script to create the appropriate database server logon for the Broker service. This can be used when configuring a mirror server for use.

# Get-BrokerDBVersionChangeScript

Jan 04, 2017

Gets an SQL service schema update script for the Citrix Broker Service.

## Syntax

```
Get-BrokerDBVersionChangeScript -DatabaseName <String> -TargetVersion <Version> [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Gets an SQL script that can be used to update the current Citrix Broker Service database schema. An update can be an upgrade or downgrade.

A script can be obtained to update the current service schema to any version that is reachable by applying available schema update packages that have been uploaded by the Citrix Broker Service.

Typically, this mechanism is used to update the current service schema to a newer version, however it can also be used to revert previously applied updates.

The SQL script obtained can be run using SQL Server's SQLCMD utility, or by copying the script into an SQL Server Management Studio (SSMS) query window and executing the query. If using SSMS, the query must be executed in 'SQLCMD mode'.

The schema update packages used to generate update scripts are stored in the database; because of this, any Citrix Broker Service in the site can be used to obtain a schema update script.

The fact that an update package is available in the database does not mean that the update has actually been applied to the service's schema. In addition, application of an update does not remove the associated update packages.

Take care when using the update scripts. Citrix recommends that where possible service schema upgrades are performed using Studio rather than manually via the SDK. The database should be backed-up before an update is attempted. The database script may also require exclusive use of the schema, in which case all Citrix Broker Services must be shutdown before applying the update.

Once an update has been applied to the service schema, any existing Citrix Broker Services that are incompatible with the updated schema will cease to operate. The service state, as reported by `Get-BrokerServiceStatus`, provides information about the service compatibility (e.g. `DBNewerVersionThanService`).

## Related topics

[Get-BrokerInstalledDBVersion](#)

[Get-BrokerServiceStatus](#)

[Get-BrokerDBSchema](#)

## Parameters

**-DatabaseName**<String>

The name of the database containing the Citrix Broker Service schema to be updated.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TargetVersion**<Version>

The required target service schema version of the update. This is the service schema version obtained after the update script is applied.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

System.Management.Automation.PSObject

The Get-BrokerDBVersionChangeScript cmdlet returns a PSObject containing a script that can be used to update the Citrix Broker Service database schema. The object has the following properties:

**-- Script**

The raw text of the SQL script to apply the update.

-- CanUndo

If true, indicates that after the update has been applied, a script to revert from the updated schema to the schema state prior to the update can be obtained. Because Get-BrokerDBVersionChangeScript gets only update scripts relating to the current schema version, a script to revert an update can be obtained only after the update has been applied.

-- NeedExclusiveAccess

If true, indicates that the update requires exclusive access to the Citrix Broker Service's schema while the update is applied; all Citrix Broker Services must be shutdown during the update.

## Notes

The PSObject returned by this cmdlet contains the following properties:

-- Script The raw text of the SQL script to apply the update, or null in the case when no upgrade path to the specified target version exists.

-- NeedExclusiveAccess Indicates whether all services in the service group must be shut down during the update or not.

-- CanUndo Indicates whether the generated script allows the updated schema to be reverted to the state prior to the update.

Scripts to update the schema version are stored in the database so any service in the service group can obtain these scripts. Extreme caution should be exercised when using update scripts. Citrix recommends backing up the database before attempting to upgrade the schema. Database update scripts may require exclusive use of the schema and so may not be able to execute while any Broker services are running. However, this depends on the specific update being carried out.

After a schema update has been carried out, services that require the previous version of the schema may cease to operate. The ServiceState parameter reported by the Get-BrokerServiceStatus command provides information about service compatibility. For example, if the schema has been upgraded to a more recent version that a service cannot use, the service reports "DBNewerVersionThanService".

If the command fails, the following errors can be returned.

## Error Codes

-----

### NoOp

The operation was successful but had no effect.

### NoDBConnections

The database connection string for the Broker Service has not been specified.

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.



## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $update = Get-BrokerDBVersionChangeScript -DatabaseName MyDb -TargetVersion 1.0.75.0
```

```
    C:\PS> $update.Script > update_75.sql
```

Gets an SQL update script to update the current schema to version 1.0.75.0. The resulting update\_75.sql script is suitable for direct use with the SQL Server SQLCMD utility.

# Get-BrokerDelayedHostingPowerAction

Jan 04, 2017

Gets power actions that are executed after a delay.

## Syntax

```
Get-BrokerDelayedHostingPowerAction [-Uid] <Int64> [-Property <String[]>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Get-BrokerDelayedHostingPowerAction [[-MachineName] <String>] [-Action <PowerManagementAction>] [-ActionDueTime  
<DateTime>] [-DNSName <String>] [-HostedMachineName <String>] [-HypervisorConnectionName <String>] [-  
HypervisorConnectionUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-  
Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Finds all delayed power actions that match the specified search criteria.

----- BrokerDelayedHostingPowerAction Object

The BrokerDelayedHostingPowerAction object represents an instance of a power action that is executed after a delay. It contains the following properties:

-- Action (Citrix.Broker.Admin.SDK.PowerManagementAction)

The power action to apply to the machine. Possible values are ShutDown and Suspend.

-- ActionDueTime (System.DateTime)

The UTC time at which the power action is due to be queued for execution.

-- DNSName (System.String)

The fully qualified DNS name of the machine that the power action applies to.

-- HostedMachineName (System.String)

The hypervisor's name for the machine that the power action applies to.

-- HypervisorConnectionUid (System.Int32)

The unique identifier of the hypervisor connection that is associated with the target machine.

-- MachineName (System.String)

The name of the machine that the power action applies to, in the form domain\machine.

-- Uid (System.Int64)

The unique identifier of the power action.

## Related topics

[New-BrokerDelayedHostingPowerAction](#)

[Remove-BrokerDelayedHostingPowerAction](#)

[Remove-BrokerHostingPowerAction](#)

#### Parameters

**-Uid**<Int64>

Gets only the single action record whose ID matches the specified value.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName**<String>

Gets only the records for actions that are for machines whose name (of the form domain\machine) matches the specified string.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Action**<PowerManagementAction>

Gets only the records for actions with the specified action type.

Valid values are Shutdown and Suspend.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ActionDueTime**<DateTime>

Gets only the records for actions due to be queued for execution at the specified time. This is useful with advanced filtering; for more information, see [about\\_Broker\\_Filtering](#).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DNSName<String>**

Gets only the records for actions that are for machines whose DNS name matches the specified string.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineName<String>**

Gets only the records for actions that are for machines whose Hosting Name (the machine name as understood by the hypervisor) matches the specified string.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionName<String>**

Gets only the records for actions for machines hosted through a hypervisor connection whose name matches the specified string.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionUid<Int32>**

Gets only the records for actions for machines hosted through a hypervisor connection whose ID matches the specified

value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See `about_Broker_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by `-ReturnTotalRecordCount`.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See `about_Broker_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|               |                                                                                        |
|---------------|----------------------------------------------------------------------------------------|
| Required?     | false                                                                                  |
| Default Value | Localhost. Once a value is provided by any cmdlet, this value will become the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

Citrix.Broker.Admin.SDK.DelayedHostingPowerAction

Get-BrokerDelayedHostingPowerAction returns all delayed power actions that match the specified selection criteria.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerDelayedHostingPowerAction
```

Fetches records for all known delayed power actions that have not yet been queued for execution.

# Get-BrokerDesktop

Jan 04, 2017

Gets desktops configured for this site.

## Syntax

```
Get-BrokerDesktop [-Uid <Int32>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerDesktop [[-MachineName <String>] [-AgentVersion <String>] [-ApplicationInUse <String>] [-AssignedClientName <String>] [-AssignedIPAddress <String>] [-AssociatedUserFullName <String>] [-AssociatedUserName <String>] [-AssociatedUserUPN <String>] [-AutonomouslyBrokered <Boolean>] [-CatalogName <String>] [-CatalogUid <Int32>] [-ClientAddress <String>] [-ClientName <String>] [-ClientVersion <String>] [-ColorDepth <ColorDepth>] [-ConnectedViaHostName <String>] [-ConnectedVialP <String>] [-ControllerDNSName <String>] [-DeliveryType <DeliveryType>] [-Description <String>] [-DesktopCondition <String>] [-DesktopGroupName <String>] [-DesktopGroupUid <Int32>] [-DesktopKind <DesktopKind>] [-DeviceId <String>] [-DNSName <String>] [-FunctionalLevel <FunctionalLevel>] [-HardwareId <String>] [-HostedMachineId <String>] [-HostedMachineName <String>] [-HostingServerName <String>] [-HypervisorConnectionName <String>] [-HypervisorConnectionUid <Int32>] [-IconUid <Int32>] [-ImageOutOfDate <Boolean>] [-InMaintenanceMode <Boolean>] [-IPAddress <String>] [-IsAssigned <Boolean>] [-IsPhysical <Boolean>] [-LastConnectionFailure <ConnectionFailureReason>] [-LastConnectionTime <DateTime>] [-LastConnectionUser <String>] [-LastDeregistrationReason <DeregistrationReason>] [-LastDeregistrationTime <DateTime>] [-LastErrorReason <String>] [-LastErrorTime <DateTime>] [-LastHostingUpdateTime <DateTime>] [-LaunchedViaHostName <String>] [-LaunchedVialP <String>] [-MachineInternalState <MachineInternalState>] [-MachineUid <Int32>] [-OSType <String>] [-OSVersion <String>] [-PersistUserChanges <PersistUserChanges>] [-PowerActionPending <Boolean>] [-PowerState <PowerState>] [-Protocol <String>] [-ProvisioningType <ProvisioningType>] [-PublishedApplication <String>] [-PublishedName <String>] [-PvdStage <PvdStage>] [-RegistrationState <RegistrationState>] [-SecureIcaActive <Boolean>] [-SecureIcaRequired <Boolean>] [-SessionHidden <Boolean>] [-SessionId <Int32>] [-SessionState <SessionState>] [-SessionStateChangeTime <DateTime>] [-SessionUid <Int64>] [-SessionUserName <String>] [-SessionUserSID <String>] [-SID <String>] [-SmartAccessTag <String>] [-StartTime <DateTime>] [-SummaryState <DesktopSummaryState>] [-Tag <String>] [-WillShutdownAfterUse <Boolean>] [-ApplicationUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet is now deprecated, please use Get-BrokerMachine.

Retrieves desktops matching the specified criteria. If no parameters are specified this cmdlet enumerates all desktops.

Get-BrokerDesktop returns objects that combine desktop configuration and state information.

For single-session desktops, session information is displayed if present. It is possible that there are more than one sessions present on single-session desktops if 'fast user switching' is enabled, this cmdlet will prefer to return information about brokered sessions (rather than, for example, unbrokered direct RDP sessions). If there is no session running, session related fields return \$null.

For multi-session desktops, no session information is ever displayed by this cmdlet, so session related fields always return \$null. Get-BrokerSession can be used to get information about sessions on both multi-session and single-session desktops.

To count desktops, rather than retrieve full details of each desktop, use Group-BrokerDesktop instead.

For information about advanced filtering options, see about\_Broker\_Filtering; for information about desktops, see about\_Broker\_Desktops.

### ----- BrokerDesktop Object

The desktop object returned represents a physical or virtual machine configured in the site that is able to run either a Microsoft Windows desktop environment, individual applications, or both.

-- AgentVersion (System.String)

Version of the Citrix Virtual Delivery Agent (VDA) installed on the desktop.

-- ApplicationsInUse (System.String[])

List of applications in use on the desktop (in the form of browser name).

-- AssignedClientName (System.String)

The name of the endpoint client device that the desktop has been assigned to.

-- AssignedIPAddress (System.String)

The IP address of the endpoint client device that the desktop has been assigned to.



-- AssociatedUserFullNames (System.String[])

Full names of the users that have been associated with the desktop (in the form "Firstname Lastname").

Associated users are the current user(s) for shared desktops and the assigned users for private desktops.

-- AssociatedUserNames (System.String[])

Usemames of the users that have been associated with the desktop (usually in the form "domain\user").

Associated users are the current user(s) for shared desktops and the assigned users for private desktops.

-- AssociatedUserUPNs (System.String[])

The user principal names of the users that have been associated with the desktop (in the form user@upndomain.com).

Associated users are the current user(s) for shared desktops and the assigned users for private desktops.

-- AutonomouslyBrokered (System.Boolean?)

Session property indicating if the current session is an HDX session established by direct connection without being brokered.

Session properties are always null for multi-session desktops.

-- CatalogName (System.String)

Name of the catalog the desktop is a member of.

-- CatalogUid (System.Int32)

UID of the catalog the desktop is a member of.

-- ClientAddress (System.String)

Session property indicating the IP address of the client connected to the desktop.

Session properties are always null for multi-session desktops.

-- ClientName (System.String)

Session property indicating the host name of the client connected to the desktop.

Session properties are always null for multi-session desktops.

-- ClientVersion (System.String)

Session property indicating the version of the Citrix Receiver running on the connected client.

Session properties are always null for multi-session desktops.

-- ColorDepth (Citrix.Broker.Admin.SDK.ColorDepth?)

The color depth setting configured on the desktop, possible values are:

Null, FourBit, EightBit, SixteenBit, and TwentyFourBit.

-- ConnectedViaHostName (System.String)

Session property indicating the host name of the connection gateway, router or client.

Session properties are always null for multi-session desktops.

-- ConnectedViaIP (System.String)

Session property indicating the IP address of the connection gateway, router or client.

Session properties are always null for multi-session desktops.

-- ControllerDNSName (System.String)

The DNS host name of the controller that the desktop is registered to.

-- DeliveryType (Citrix.Broker.Admin.SDK.DeliveryType)  
Denotes whether the desktop delivers desktops only, apps only or both.

-- Description (System.String)  
Description of the desktop.

-- DesktopConditions (System.String[])  
List of outstanding desktop conditions for the desktop.

-- DesktopGroupName (System.String)  
Name of the desktop group the desktop has been assigned to.

-- DesktopGroupUid (System.Int32)  
Uid of the desktop group the desktop has been assigned to.

-- DesktopKind (Citrix.Broker.Admin.SDK.DesktopKind)  
Deprecated.  
Denotes whether the desktop is private or shared. AllocationType should be used instead.

-- DeviceId (System.String)  
Session property indicating a unique identifier for the client device that has most recently been associated with the current session.  
Session properties are always null for multi-session desktops.

-- DNSName (System.String)  
The DNS host name of the desktop.

-- FunctionalLevel (Citrix.Broker.Admin.SDK.FunctionalLevel?)  
The functional level of the desktop, if known.

-- HardwareId (System.String)  
Session property indicating a unique identifier for the client hardware that has been most recently associated with the current session.  
Session properties are always null for multi-session desktops.

-- HostedMachineId (System.String)  
Unique ID within the hosting unit of the target managed desktop.

-- HostedMachineName (System.String)  
The friendly name of a hosted desktop as used by its hypervisor. This is not necessarily the DNS name of the desktop.

-- HostingServerName (System.String)  
DNS name of the hypervisor that is hosting the desktop if managed.

-- HypervisorConnectionName (System.String)  
The name of the hypervisor connection that the desktop's hosting server is accessed through, if managed.

-- HypervisorConnectionUid (System.Int32?)  
The UID of the hypervisor connection that the desktop's hosting server is accessed through, if managed.

-- IconUid (System.Int32?)  
The UID of the desktop's icon that is displayed in StoreFront.

-- ImageOutOfDate (System.Boolean?)

Denotes whether the VM image for a hosted desktop is out of date.

-- InMaintenanceMode (System.Boolean)

Denotes whether the desktop is in maintenance mode.

-- IPAddress (System.String)

The IP address of the desktop.

-- IsAssigned (System.Boolean)

Denotes whether a private desktop has been assigned to a user/users, or a client name/address. Users can be assigned explicitly or by assigning on first use of the desktop.

-- IsPhysical (System.Boolean)

This value is true if the desktop is physical (ie not power managed by the Citrix Broker Service), and false otherwise.

-- LastConnectionFailure (Citrix.Broker.Admin.SDK.ConnectionFailureReason)

The reason for the last failed connection between a client and the desktop.

-- LastConnectionTime (System.DateTime?)

Time of the last detected connection attempt that either failed or succeeded.

-- LastConnectionUser (System.String)

The SAM name (in the form DOMAIN\user) of the user that last attempted a connection with the desktop. If the SAM name is not available, the SID is used.

-- LastDeregistrationReason (Citrix.Broker.Admin.SDK.DeregistrationReason?)

The reason for the last deregistration of the desktop with the broker. Possible values are:

AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

-- LastDeregistrationTime (System.DateTime?)

Time of the last deregistration of the desktop from the controller.

-- LastErrorReason (System.String)

The reason for the last error detected in the desktop.

-- LastErrorTime (System.DateTime?)

The time of the last detected error.

-- LastHostingUpdateTime (System.DateTime?)

Time of last update to any hosting data (such as power state) for this desktop reported by the hypervisor connection.

-- LaunchedViaHostName (System.String)

Session property that denotes the host name of the StoreFront server used to launch the current brokered session.

Session properties are always null for multi-session desktops.

-- LaunchedViaIP (System.String)

Session property that denotes the IP address of the StoreFront server used to launch the current brokered session.

Session properties are always null for multi-session desktops.

-- MachineInternalState (Citrix.Broker.Admin.SDK.MachineInternalState)

The internal state of the machine associated with the desktop; reported while the desktop is registered to a controller, plus some private Citrix Broker Service states while the machine is not registered.

-- MachineName (System.String)

DNS host name of the machine associated with the desktop.

-- MachineUid (System.Int32)

Uid of the associated machine.

-- OSType (System.String)

A string that can be used to identify the operating system that is running on the desktop.

-- OSVersion (System.String)

A string that can be used to identify the version of the operating system running on the desktop, if known

-- PersistUserChanges (Citrix.Broker.Admin.SDK.PersistUserChanges)

Describes whether/how the user changes are persisted. Possible values are:

o OnLocal - Persist the user changes on the local disk of the desktop.

o Discard - Discard user changes.

o OnPvd - Persist user changes on the Citrix Personal vDisk.

-- PowerActionPending (System.Boolean)

Property indicating whether there are any pending power actions for the desktop.

-- PowerState (Citrix.Broker.Admin.SDK.PowerState)

The current power state of the desktop. Possible values are: Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, resuming.

-- Protocol (System.String)

Session property that denotes the protocol that the current session is using, can be either HDX, RDP or Console. Console sessions on XenDesktop 5 VDAs appear with a blank protocol.

Session properties are always null for multi-session desktops.

-- ProvisioningType (Citrix.Broker.Admin.SDK.ProvisioningType)

Describes how the machine associated with the desktop was provisioned, possible values are:

o Manual: No automated provisioning.

o PVS: Machine provisioned by PVS (may be physical, blade, VM,...)

o MCS: Machine provisioned by MCS (machine must be VM)

-- PublishedApplications (System.String[])

List of applications published by the desktop (displayed as browser names).

-- PublishedName (System.String)

The name of the desktop that is displayed in StoreFront, if the desktop is published.

-- PvdStage (Citrix.Broker.Admin.SDK.PvdStage)

For a desktop supporting Personal vDisk technology (PvD), indicates the stage of the PvD image preparation.

-- RegistrationState (Citrix.Broker.Admin.SDK.RegistrationState)

Indicates the registration state of the desktop. Possible values are: Unregistered, Initializing, Registered, AgentError.

-- SecureIcaActive (System.Boolean?)

Session property that indicates whether SecureICA is active on the current session.

Session properties are always null for multi-session desktops.

-- SecureIcaRequired (System.Boolean?)

Flag indicating whether SecureICA is required or not when starting a session on the desktop.

-- SessionHidden (System.Boolean?)

Session property that indicates if a session is hidden.

Session properties are always null for multi-session desktops.

-- SessionId (System.Int32?)

Deprecated. A unique identifier that Remote Desktop Services uses to track the session but it is only unique on that machine and only unique at any one particular time.

-- SessionState (Citrix.Broker.Admin.SDK.SessionState?)

Session property indicating the state of the current session.

Session properties are always null for multi-session desktops, possible values are: Other, PreparingSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession and Unknown.

-- SessionStateChangeTime (System.DateTime?)

Session property indicating the time of the last state change of the current session.

Session properties are always null for multi-session desktops.

-- SessionUid (System.Int64?)

Session property indicating the UID of the current session.

Session properties are always null for multi-session desktops.

-- SessionUserName (System.String)

Session property indicates the name of the current sessions' user (in the form DOMAIN\user).

Session properties are always null for multi-session desktops.

-- SessionUserSID (System.String)

Session property indicates the SID of the current sessions' user.

Session properties are always null for multi-session desktops.

-- SID (System.String)

The SID of the desktop.

-- SmartAccessTags (System.String[])

Session property that indicates the Smart Access tags for the current session.

Session properties are always null on multi-session desktops.

-- StartTime (System.DateTime?)

Session property that indicates the start time of the current session.

Session properties are always null on multi-session desktops.

-- SummaryState (Citrix.Broker.Admin.SDK.DesktopSummaryState)

Indicates the overall state of the desktop. The overall state is a result of other more specific states such as session state, registration state and power state. Possible values: Off, Unregistered, Available, Disconnected, InUse, Preparing.

-- Tags (System.String[])

A list of tags for the desktop.

-- Uid (System.Int32)

UID of the desktop object.

-- WillShutdownAfterUse (System.Boolean)

Flag indicating whether this desktop is tainted and will be shut down after all sessions on the desktop have ended. This flag should only ever be true on power managed, single-session desktops.

Note: The desktop will not shut down if it is in maintenance mode, but will shut down after the desktop is taken out of maintenance mode.

Related topics

[Group-BrokerMachine](#)

[Get-BrokerMachine](#)

Parameters

**-Uid**<Int32>

Gets desktops with a specific UID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName**<String>

Gets desktops with a specific machine name (in the form 'domain\machine').

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AgentVersion**<String>

Gets desktops with a specific Citrix Virtual Delivery Agent version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationInUse**<String>

Gets desktops running a specified published application (identified by browser name).

String comparisons are case-insensitive.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-AssignedClientName<String>**

Gets desktops assigned to a specific client name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedIPAddress<String>**

Gets desktops assigned to a specific client IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserFullName<String>**

Gets desktops with an associated user identified by their full name (usually in the form 'first-name last-name').

Associated users are the current user for shared desktops, and the assigned users for private desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserName<String>**

Gets desktops with an associated user identified by their user name (in the form 'domain\user').

Associated users are the current user for shared desktops, and the assigned users for private desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserUPN<String>**

Gets desktops with an associated user identified by their User Principle Name (in the form 'user@domain').

Associated users are the current user for shared desktops, and the assigned users for private desktops.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AutonomouslyBrokered**<Boolean>

Gets desktops according to whether their current session is autonomously brokered or not. Autonomously brokered sessions are HDX sessions established by direct connection without being brokered.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CatalogName**<String>

Gets desktops from the catalog with the specific name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CatalogUid**<Int32>

Gets desktops from a catalog with a specific UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ClientAddress**<String>

Gets desktops with a specific client IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ClientName**<String>

Gets desktops with a specific client name.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |



|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ClientVersion**<String>

Gets desktops with a specific client version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ColorDepth**<ColorDepth>

Gets desktops configured with a specific color depth.

Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ConnectedViaHostName**<String>

Gets desktops with a specific host name of the incoming connection. This is usually a proxy or Citrix Access Gateway server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ConnectedViaIP**<String>

Gets desktops with a specific IP address of the incoming connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ControllerDNSName**<String>

Gets desktops with a specific DNS name of the controller they are registered with.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DeliveryType**<DeliveryType>

Gets desktops of a particular delivery type.

Valid values are AppsOnly, DesktopsOnly, DesktopsAndApps

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Gets desktops with a specific description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopCondition**<String>

Gets desktop with an outstanding desktop condition condition.

Valid values are:

- o CPU: Indicates the machine has high CPU usage
- o ICALatency: Indicates the network latency is high
- o UPMLogonTime: Indicates that the profile load time was high

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupName**<String>

Gets desktops from a desktop group with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Gets desktops from a desktop group with the specified UID.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopKind**<DesktopKind>

Deprecated: Use AllocationType parameter.

Gets desktops of a particular kind.

Valid values are Private, Shared.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DeviceId**<String>

Gets desktops with a specific client device ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DNSName**<String>

Gets desktops with a specific DNS name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-FunctionalLevel**<FunctionalLevel>

Gets desktops with a specific FunctionalLevel.

Valid values are L5, L7, L7\_6

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HardwareId**<String>

Gets desktops with a specific client hardware ID.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineId**<String>

Gets desktops with a specific machine ID known to the hypervisor.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineName**<String>

Gets desktops with a specific machine name known to the hypervisor.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostingServerName**<String>

Gets desktops with a specific name of the hosting hypervisor server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionName**<String>

Gets desktops with a specific name of the hosting hypervisor connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionUid**<Int32>

Gets desktops with a specific UID of the hosting hypervisor connection.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-IconUid<Int32>**

Gets desktops with a specific configured icon. Note that desktops with a null IconUid use the icon of the desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ImageOutOfDate<Boolean>**

Gets desktops by whether their disk image is out of date (for machines provisioned using MCS only).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InMaintenanceMode<Boolean>**

Gets desktops with a specific InMaintenanceMode setting.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IPAddress<String>**

Gets desktops with a specific IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsAssigned<Boolean>**

Gets desktops according to whether they are assigned or not. Desktops may be assigned to one or more users or groups, a client IP address or a client endpoint name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsPhysical**<Boolean>

Specifies if machines in the catalog can be power managed by the Citrix Broker Service. Where the power state of the machine cannot be controlled, specify `True`, otherwise `False`. Can only be specified together with a provisioning type of `Pvs` or `Manual`, or if used with the deprecated `CatalogKind` parameter only with `Pvs` or `PvsPvd` catalog kinds.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastConnectionFailure**<ConnectionFailureReason>

Gets desktops with a specific reason for the last recorded connection failure. This value is `None` if the last connection was successful or if there has been no attempt to connect to the desktop yet.

Valid values are `None`, `SessionPreparation`, `RegistrationTimeout`, `ConnectionTimeout`, `Licensing`, `Ticketing`, and `Other`.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastConnectionTime**<DateTime>

Gets desktops that last connected at a specific time. This is the time that the broker detected that the connection attempt either succeeded or failed.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastConnectionUser**<String>

Gets desktops where a specific user name last attempted a connection (in the form 'domain\user').

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastDeregistrationReason**<DeregistrationReason>

Gets desktops whose broker last recorded a specific deregistration reason.

Valid values are `$null`, `AgentShutdown`, `AgentSuspended`, `AgentRequested`, `IncompatibleVersion`, `AgentAddressResolutionFailed`, `AgentNotContactable`, `AgentWrongActiveDirectoryOU`, `EmptyRegistrationRequest`, `MissingRegistrationCapabilities`, `MissingAgentVersion`, `InconsistentRegistrationCapabilities`, `NotLicensedForFeature`, `UnsupportedCredentialSecurityVersion`, `InvalidRegistrationRequest`, `SingleMultiSessionMismatch`, `FunctionalLevelTooLowForCatalog`, `FunctionalLevelTooLowForDesktopGroup`, `PowerOff`, `DesktopRestart`, `DesktopRemoved`, `AgentRejectedSettingsUpdate`, `SendSettingsFailure`, `SessionAuditFailure`, `SessionPrepareFailure`, `ContactLost`, `SettingsCreationFailure`, `UnknownError` and `BrokerRegistrationLimitReached`.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastDeregistrationTime**<DateTime>

Gets desktops by the time that they were last deregistered.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastErrorReason**<String>

Gets desktops with the specified last error reason.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastErrorTime**<DateTime>

Gets desktops with the specified last error time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastHostingUpdateTime**<DateTime>

Gets desktops with a specific time that the hosting information was last updated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LaunchedViaHostName**<String>

Gets desktops with a specific host name of the StoreFront server from which the user launched the session.

Session properties are always null for multi-session desktops.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-LaunchedViaIP**<String>

Gets desktops with a specific IP address of the StoreFront server from which the user launched the session.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineInternalState**<MachineInternalState>

Gets desktops with the specified internal machine state.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineUid**<Int32>

Gets desktops with a specific machine UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSType**<String>

Gets desktops by the type of operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSVersion**<String>

Gets desktops by the version of the operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-PersistUserChanges**<PersistUserChanges>

Gets desktops by the location where the user changes are persisted.

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the Pvd.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PowerActionPending**<Boolean>

Gets desktops with a specific power action pending state.

Valid values are \$true or \$false.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PowerState**<PowerState>

Gets desktops with a specific power state.

Valid values are Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Protocol**<String>

Gets desktops with connections using a specific protocol, for example HDX, RDP, or Console.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ProvisioningType**<ProvisioningType>

Gets desktops that are in a catalog with a particular provisioning type. Values can be:

- o Manual - No provisioning.

o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).

o MCS - Machine provisioned by MCS (machine must be VM).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedApplication**<String>

Gets desktops with a specific application published to them.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName**<String>

Gets desktops with a specific published name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvdStage**<PvdStage>

Gets desktops with a specific personal vDisk stage.

Valid values are None, Requested, Starting, Working and Failed.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RegistrationState**<RegistrationState>

Gets desktops with a specific registration state.

Valid values are Unregistered, Initializing, Registered and AgentError.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaActive**<Boolean>

Gets desktops depending on whether the current session uses SecureICA or not.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaRequired**<Boolean>

Gets desktops configured with a particular SecureIcaRequired setting. Note that the desktop setting of \$null indicates that the desktop group value is used.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionHidden**<Boolean>

Gets desktops by whether their sessions are hidden or not. Hidden sessions are treated as though they do not exist when launching sessions; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionId**<Int32>

Deprecated.

Gets desktops by session ID, a unique identifier that Remote Desktop Services uses to track the session but it is only unique on that machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionState**<SessionState>

Gets desktops with a specific session state.

Valid values are \$null, Other, PreparingSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession, and Unknown.

Session properties are always null for multi-session desktops.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionStateChangeTime**<DateTime>

Gets desktops whose sessions last changed state at a specific time.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUid**<Int64>

Gets single-session desktops with a specific session UID (\$null for no session).

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUserName**<String>

Gets desktops with a specific user name for the current session (in the form 'domain\user').

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUserSID**<String>

Gets desktops with a specific SID of the current session user.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SID**<String>

Gets desktops with a specific machine SID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SmartAccessTag**<String>

Gets session desktops where the session has the specific SmartAccess tag.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartTime**<DateTime>

Gets desktops with a specific session start time.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SummaryState**<DesktopSummaryState>

Gets desktops with a specific summary state.

Valid values are Off, Unregistered, Available, Disconnected, InUse and Preparing.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Tag**<String>

Gets desktops with a specific tag.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WillShutdownAfterUse**<Boolean>

Gets desktops depending on whether they shut down after use or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationUid**<Int32>

Gets desktops with a specific published application (identified by its UID).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each

name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.Desktop

Get-BrokerDesktop returns an object for each matching desktop.

**Notes**

To compare dates or times, use -Filter and relative comparisons. For more information, see about\_Broker\_Filtering and the examples.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerDesktop -RegistrationState Unregistered
```

```
C:\PS> Get-BrokerDesktop -Filter { RegistrationState -ne 'Registered' }
```

Both commands retrieve desktops that are unregistered. The second command also includes desktops with a registration state of AgentError.

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerDesktop -SessionUid $null | ft -a DNSName,SummaryState
```

Gets desktops without sessions, listing the DNS name and current state.

----- **EXAMPLE 3** -----

```
C:\PS> Get-BrokerDesktop -Filter { OSType -like "Windows XP*" -and ImageOutOfDate }
```

Finds all Windows XP desktops with an out-of-date image.

----- **EXAMPLE 4** -----

```
C:\PS> Get-BrokerDesktop -ApplicationInUse '*powerpoint*
```

Gets desktops running a published PowerPoint application. It matches any application browser name containing the word 'powerpoint'. String comparisons are case-insensitive.

----- **EXAMPLE 5** -----

```
C:\PS> Get-BrokerDesktop -DesktopCondition * DNSName,SessionUserName,DesktopConditions
```

Finds all desktops with an outstanding desktop condition, listing the affected desktop and user.

----- **EXAMPLE 6** -----

```
C:\PS> $d = (Get-Date).AddDays(-1)
```

```
C:\PS> Get-BrokerDesktop -Filter { StartTime -le $d } | ft MachineName,SessionUserName,StartTime,@{Label='Duration'; Expression={(Get-Date) - $_.StartTime}}
```

Finds users who have been logged on for more than a day, and outputs the machine name, start time, and duration the session has been logged on.



# Get-BrokerDesktopGroup

Jan 04, 2017

Gets broker desktop groups configured for this site.

## Syntax

```
Get-BrokerDesktopGroup [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerDesktopGroup [[-Name] <String>] [-AutomaticPowerOnForAssigned <Boolean>] [-AutomaticPowerOnForAssignedDuringPeak <Boolean>] [-ColorDepth <ColorDepth>] [-DeliveryType <DeliveryType>] [-Description <String>] [-DesktopKind <DesktopKind>] [-Enabled <Boolean>] [-InMaintenanceMode <Boolean>] [-IsRemotePC <Boolean>] [-Metadata <String>] [-MinimumFunctionalLevel <FunctionalLevel>] [-OffPeakBufferSizePercent <Int32>] [-OffPeakDisconnectAction <SessionChangeHostingAction>] [-OffPeakDisconnectTimeout <Int32>] [-OffPeakExtendedDisconnectAction <SessionChangeHostingAction>] [-OffPeakExtendedDisconnectTimeout <Int32>] [-OffPeakLogOffAction <SessionChangeHostingAction>] [-OffPeakLogOffTimeout <Int32>] [-PeakBufferSizePercent <Int32>] [-PeakDisconnectAction <SessionChangeHostingAction>] [-PeakDisconnectTimeout <Int32>] [-PeakExtendedDisconnectAction <SessionChangeHostingAction>] [-PeakExtendedDisconnectTimeout <Int32>] [-PeakLogOffAction <SessionChangeHostingAction>] [-PeakLogOffTimeout <Int32>] [-PublishedName <String>] [-Scopeld <Guid>] [-ScopeName <String>] [-SecureIcaRequired <Boolean>] [-SessionSupport <SessionSupport>] [-SettlementPeriodBeforeAutoShutdown <TimeSpan>] [-ShutdownDesktopsAfterUse <Boolean>] [-Tag <String>] [-TimeZone <String>] [-TotalApplications <Int32>] [-TurnOnAddedMachine <Boolean>] [-UUID <Guid>] [-ApplicationUid <Int32>] [-TagUid <Int32>] [-PowerTimeSchemeUid <Int32>] [-MachineConfigurationUid <Int32>] [-RemotePCCatalogUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieve desktop groups matching the specified criteria. If no parameters are specified this cmdlet enumerates all desktop groups.

Desktop groups represent groups of desktops that are managed together for brokering purposes.

----- BrokerDesktopGroup Object

A desktop group object represents a collection of machines that are fully configured in a site that is able to run either a Microsoft Windows desktop environment, individual applications, or both.

-- AutomaticPowerOnForAssigned (System.Boolean)

Specifies whether assigned desktops in the desktop group are automatically started at the start of peak time periods. Only relevant for groups whose DesktopKind is Private.

-- AutomaticPowerOnForAssignedDuringPeak (System.Boolean)

Specifies whether assigned desktops in the desktop are automatically started throughout peak time periods. Only relevant for groups whose DesktopKind is Private and which have AutomaticPowerOnForAssigned set to true.

-- ColorDepth (Citrix.Broker.Admin.SDK.ColorDepth)

Default color depth of sessions started with machines in the desktop group. Possible values are:

FourBit, EightBit, SixteenBit, TwentyFourBit.

-- ConfigurationSlotUids (System.Int32[])

Uids of any configuration slots which hold machine configurations associated with the desktop group. The order of slot UIDs in this list correspond with the order of items in the associated MachineConfigurationNames and MachineConfigurationUids list properties, and so the same slot UID can appear more than once.

-- DeliveryType (Citrix.Broker.Admin.SDK.DeliveryType)

The type of resources being published. Possible values are:

DesktopsOnly, AppsOnly, DesktopsAndApps.

-- Description (System.String)

Description of the desktop group.

-- DesktopKind (Citrix.Broker.Admin.SDK.DesktopKind)

The kind of the desktops being published, possible values are:

Private and Shared.

-- DesktopsAvailable (System.Int32)

The number of machines in the desktop group in state Available; this is the number of machines with no sessions present.

-- DesktopsDisconnected (System.Int32)

The number of disconnected sessions present on machines in the desktop group.

-- DesktopsInUse (System.Int32)

The number of machines in the desktop group in state InUse; this is the number of machines with at least one session present.

-- DesktopsNeverRegistered (System.Int32)

The number of machines in the desktop group that have never registered with the current site.

-- DesktopsPreparing (System.Int32)

The number of machines in the desktop group whose PvD disk image is being prepared.

-- DesktopsUnregistered (System.Int32)

The number of machines in the desktop group that are currently unregistered.

-- Enabled (System.Boolean)

Specifies whether the desktop group is enabled or not; disabled desktop groups do not appear to users.

-- IconUid (System.Int32)

The Uid of the icon to be used as a default for desktops in the desktop group. Individual desktop objects can override this default by setting the IconUid parameter on the desktop object.

-- InMaintenanceMode (System.Boolean)

Specifies whether the machines in the desktop group are in maintenance mode or not.

-- IsRemotePC (System.Boolean)

Specifies whether the desktop group is a Remote PC desktop group.

-- MachineConfigurationNames (System.String[])

The MachineConfiguration names associated with the desktop group.

-- MachineConfigurationUids (System.Int32[])

The MachineConfiguration uids associated with the desktop group.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Metadata associated with the desktop group.

-- MinimumFunctionalLevel (Citrix.Broker.Admin.SDK.FunctionalLevel)

The minimum FunctionalLevel required for the machines in the desktop group to be able to register with the Citrix Broker Service.

-- Name (System.String)

Name of the desktop group.

-- OffPeakBufferSizePercent (System.Int32)

The percentage of machines that are kept available in an idle state outside peak hours.

-- OffPeakDisconnectAction (Citrix.Broker.Admin.SDK.SessionChangeHostingAction)

The action that is performed after a configurable period of a user session disconnecting outside peak hours. Possible values are Nothing, Suspend or Shutdown.

-- OffPeakDisconnectTimeout (System.Int32)

The number of minutes before the configured action is performed after a user session disconnects outside peak hours.

-- OffPeakExtendedDisconnectAction (Citrix.Broker.Admin.SDK.SessionChangeHostingAction)

The action performed after a second configurable period of a user session disconnecting outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

-- OffPeakExtendedDisconnectTimeout (System.Int32)

The number of minutes before the second configured action is performed after a user session disconnects outside peak hours.

-- OffPeakLogOffAction (Citrix.Broker.Admin.SDK.SessionChangeHostingAction)

The action performed after a configurable period of a user session ending outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

-- OffPeakLogOffTimeout (System.Int32)

The number of minutes before the configured action is performed after a user session ends outside peak hours.

-- PeakBufferSizePercent (System.Int32)

The percentage of machines in the desktop group that are kept available in an idle state in peak hours.

-- PeakDisconnectAction (Citrix.Broker.Admin.SDK.SessionChangeHostingAction)

The action performed after a configurable period of a user session disconnecting in peak hours. Possible values are Nothing, Suspend, or Shutdown.

-- PeakDisconnectTimeout (System.Int32)

The number of minutes before the configured action is performed after a user session disconnects in peak hours.

-- PeakExtendedDisconnectAction (Citrix.Broker.Admin.SDK.SessionChangeHostingAction)

The action performed after a second configurable period of a user session disconnecting in peak hours. Possible values are Nothing, Suspend, or Shutdown.

-- PeakExtendedDisconnectTimeout (System.Int32)

The number of minutes before the second configured action is performed after a user session disconnects in peak hours.

-- PeakLogOffAction (Citrix.Broker.Admin.SDK.SessionChangeHostingAction)

The action performed after a configurable period of a user session ending in peak hours. Possible values are Nothing, Suspend, or Shutdown.

-- PeakLogOffTimeout (System.Int32)

The number of minutes before the configured action is performed after a user session ends in peak hours.

-- ProtocolPriority (System.String[])

A list of protocol names in the order in which they are attempted for use during connection.

-- PublishedName (System.String)

The name of the desktop group as it is to appear to the user in StoreFront.

-- Scopes (Citrix.Broker.Admin.SDK.ScopeReference[])

The list of the delegated admin scopes to which the desktop group belongs.

-- SecureIcaRequired (System.Boolean)

Flag that specifies if the SecureICA encryption of the HDX protocol is required for sessions of desktops in the desktop

group.

-- Sessions (System.Int32)

The total number of user sessions currently running on all of the machines in the desktop group.

-- SessionSupport (Citrix.Broker.Admin.SDK.SessionSupport)

Specifies the session support (single/multi) of the machines in the desktop group. Machines with the incorrect session support for the desktop group will be unable to register with the Citrix Broker Service.

-- SettlementPeriodBeforeAutoShutdown (System.TimeSpan)

Time after a session ends during which automatic shutdown requests (for example, shutdown after use, idle pool management) are deferred. Any outstanding shutdown request takes effect after the settlement period expires. This is typically used to configure time to allow logoff scripts to complete.

-- ShutdownDesktopsAfterUse (System.Boolean)

Specifies if the desktops will shut down after they have been used and there are no sessions running on the machine. The machines will not shut down if they are placed into maintenance mode, even if this flag is set to \$true. The machines, however, will shutdown after the machine is taken out of maintenance mode if the flag is still set.

-- Tags (System.String[])

Tags associated with the desktop group.

-- TimeZone (System.String)

The timezone that desktops in the desktop group are in (for power policy purposes).

-- TotalApplications (System.Int32)

Total number of applications associated with the desktop group.

-- TotalDesktops (System.Int32)

Total number of machines in the desktop group.

-- TurnOnAddedMachine (System.Boolean)

Specifies whether the broker should attempt to turn on power-managed machines when they are added to the desktop group.

-- Uid (System.Int32)

Uid of the desktop group.

-- UUID (System.Guid)

UUID of the desktop group.

Related topics

[New-BrokerDesktopGroup](#)

[Set-BrokerDesktopGroup](#)

[Rename-BrokerDesktopGroup](#)

[Remove-BrokerDesktopGroup](#)

[Add-BrokerUser](#)

[Add-BrokerTag](#)

Parameters

**-Uid**<Int32>

Gets desktop groups with the specified value of Uid.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Name**<String>

Gets desktop groups whose name matches the supplied pattern.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AutomaticPowerOnForAssigned**<Boolean>

Gets only desktop groups with the specified value of AutomaticPowerOnForAssigned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AutomaticPowerOnForAssignedDuringPeak**<Boolean>

Gets only desktop groups with the specified value of AutomaticPowerOnForAssignedDuringPeak.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ColorDepth**<ColorDepth>

Gets only desktop groups with the specified color depth.

Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DeliveryType**<DeliveryType>

Gets desktop groups according to their delivery type.

Valid values are DesktopsOnly, AppsOnly and DesktopsAndApps.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Gets desktop groups whose description matches the supplied pattern.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopKind**<DesktopKind>

Gets desktops of a particular kind.

Valid values are Private and Shared.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Gets desktop groups with the specified value of Enabled.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InMaintenanceMode**<Boolean>

Gets desktop groups with the specified value of InMaintenanceMode.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsRemotePC**<Boolean>

Gets desktop groups with the specified IsRemotePC value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>



Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MinimumFunctionalLevel**<FunctionalLevel>

Gets desktop groups with a specific MinimumFunctionalLevel.

Valid values are L5, L7, L7\_6

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OffPeakBufferSizePercent**<Int32>

Gets desktop groups with the specified value of OffPeakBufferSizePercent.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OffPeakDisconnectAction**<SessionChangeHostingAction>

Gets desktop groups with the specified value of OffPeakDisconnectAction.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

---

**-OffPeakDisconnectTimeout**<Int32>

Gets desktop groups with the specified value of OffPeakDisconnectTimeout.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OffPeakExtendedDisconnectAction**<SessionChangeHostingAction>

Gets desktop groups with the specified value of OffPeakExtendedDisconnectAction.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OffPeakExtendedDisconnectTimeout**<Int32>

Gets desktop groups with the specified value of OffPeakExtendedDisconnectTimeout.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OffPeakLogOffAction**<SessionChangeHostingAction>

Gets desktop groups with the specified value of OffPeakLogOffAction.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OffPeakLogOffTimeout**<Int32>

Gets desktop groups with the specified value of OffPeakLogOffTimeout.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakBufferSizePercent**<Int32>

Gets desktop groups with the specified value of PeakBufferSizePercent.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakDisconnectAction**<SessionChangeHostingAction>

Gets desktop groups with the specified value of PeakDisconnectAction.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakDisconnectTimeout**<Int32>

Gets desktop groups with the specified value of PeakDisconnectTimeout.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakExtendedDisconnectAction**<SessionChangeHostingAction>

Gets desktop groups with the specified value of PeakExtendedDisconnectAction.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakExtendedDisconnectTimeout<Int32>**

Gets desktop groups with the specified value of PeakExtendedDisconnectTimeout.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakLogOffAction<SessionChangeHostingAction>**

Gets desktop groups with the specified value of PeakLogOffAction.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakLogOffTimeout<Int32>**

Gets desktop groups with the specified value of PeakLogOffTimeout.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName<String>**

Gets desktop groups whose published name matches the supplied pattern.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ScopeId**<Guid>

Gets desktop groups that are associated with the given scope identifier.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ScopeName**<String>

Gets desktop groups that are associated with the given scope name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaRequired**<Boolean>

Gets desktop groups with the specified value of SecureIcaRequired.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionSupport**<SessionSupport>

Gets desktop groups that have the specified session capability. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SettlementPeriodBeforeAutoShutdown**<TimeSpan>

Gets desktop groups with the specified value of SettlementPeriodBeforeAutoShutdown.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ShutdownDesktopsAfterUse**<Boolean>

Gets desktop groups with the specified value of ShutdownDesktopsAfterUse.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Tag**<String>

Gets desktop groups tagged with the specified tag.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TimeZone**<String>

Gets desktop groups with the specified value of TimeZone.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TotalApplications**<Int32>

Gets desktop groups that are acting as delivery groups for the specified number of applications.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TurnOnAddedMachine**<Boolean>

Gets desktop groups with the specified value of TurnOnAddedMachine value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UUID**<Guid>

Gets desktop groups with the specified value of UUID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationUid**<Int32>

Gets desktop groups that publish the specified application (identified by Uid)

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-TagUid<Int32>**

Gets desktop groups to which the specified tag (identified by its Uid) has been added to help identify it - see Add-BrokerTag for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PowerTimeSchemeUid<Int32>**

Gets desktop groups associated with the specified power time scheme (identified by its Uid).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineConfigurationUid<Int32>**

Gets desktop groups with the specified value of MachineConfiguration.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemotePCCatalogUid<Int32>**

Gets Remote PC desktop groups associated with the specified catalog.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |



|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-Sort By**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See `about_Broker_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.DesktopGroup

Get-BrokerDesktopGroup returns an object for each matching desktop group.

## Notes

To perform greater-than or less-than comparisons, use -Filter. For more information, see about\_Broker\_Filtering and the examples.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerDesktopGroup -PublishedName EMEA*  
Finds all desktop groups with published names starting with "EMEA".
```

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerDesktopGroup -InMaintenanceMode $true  
Finds all desktop groups in maintenance mode.
```

# Get-BrokerDesktopUsage

Jan 04, 2017

Get usage history of desktop groups.

## Syntax

```
Get-BrokerDesktopUsage [-DesktopGroupName <String>] [-DesktopGroupUid <Int32>] [-InUse <Int32>] [-Timestamp <DateTime>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns information, recorded by the broker on an hourly basis, about the number of desktops in use for each desktop group. Analyzing the historical usage records can give some guidance on usage patterns and help choosing idle pool settings.

Without parameters, Get-BrokerDesktopUsage returns the first 250 records. By using parameters, you can be more selective about the records that are returned.

To retrieve more than the default 250 records, use the -MaxRecordCount parameter. To select data for a specific desktop group, use either the -DesktopGroupName or -DesktopGroupUid parameters.

See the examples for this cmdlet and about\_Broker\_Filtering for details of how to perform advanced filtering.

----- BrokerDesktopUsage Object

Desktop usage object contains information to tell how many desktops in a desktop group are in use at a given time (identified by a timestamp).

-- DesktopGroupUid (System.Int32)

Uid of the desktop group that the usage data corresponds to.

-- InUse (System.Int32)

Specifies how many desktop are in use at the time the timestamp corresponds to.

-- Timestamp (System.DateTime)

Date and time the desktop usage information corresponds to.

## Related topics

### Parameters

**-DesktopGroupName**<String>

Gets usage records for the named desktop group or for multiple desktop groups if wildcards have been specified.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Gets usage records for a specific desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InUse**<Int32>

Gets usage records where the in-use count matches the specified value. This is useful when checking for zero or when used inside a -Filter expression.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Timestamp**<DateTime>

Gets usage records that occurred at the given time.

In general, Citrix recommends, using -Filter and relative comparisons. For a demonstration, see the examples.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None You cannot pipe objects to this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.DesktopUsage

Get-BrokerDesktopUsage returns an object for each matching record.

Notes

Desktop usage information is automatically deleted after 7 days.

Examples

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerDesktopUsage -DesktopGroupName TestGroup -MaxRecordCount 24 -SortBy 'Timestamp' | ft -a @{Name='Time';Expression='{0:t}' -f \$\_.Timestamp},InUse  
Returns the last 24 hours of usage information for desktop group TestGroup, formatting it as two columns labeled Time and InUse.

----- **EXAMPLE 2** -----

C:\PS> \$d = Get-Date -Hour 0 -Minute 0 -Second 0  
C:\PS> Get-BrokerDesktopUsage -Filter { DesktopGroupName -eq 'TestGroup' -and Timestamp -ge \$d }  
Returns today's usage information for desktop group TestGroup.

----- **EXAMPLE 3** -----

C:\PS> \$dg = Get-BrokerDesktopGroup TestGroup  
C:\PS> Get-BrokerDesktopUsage -DesktopGroupUid \$dg.Uid | Select Timestamp,InUse,@{Name='Percent';Expression='{0:P0}' -f (\$\_.InUse / \$dg.TotalDesktops)}}  
Retrieves the usage history for desktop group TestGroup and adds a column showing the number of desktops in that group in use, as a percentage.

# Get-BrokerEntitlementPolicyRule

Jan 04, 2017

Gets desktop rules from the site's entitlement policy.

## Syntax

```
Get-BrokerEntitlementPolicyRule [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerEntitlementPolicyRule [[-Name] <String>] [-BrowserName <String>] [-ColorDepth <ColorDepth>] [-Description <String>] [-DesktopGroupUid <Int32>] [-Enabled <Boolean>] [-ExcludedUser <User>] [-ExcludedUserFilterEnabled <Boolean>] [-IconUid <Int32>] [-IncludedUser <User>] [-IncludedUserFilterEnabled <Boolean>] [-Metadata <String>] [-PublishedName <String>] [-SecureIcaRequired <Boolean>] [-SessionReconnection <SessionReconnection>] [-UUID <Guid>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns desktop rules matching the specified search criteria from the site's entitlement policy. If no search criteria are specified, all desktop rules in the entitlement policy are obtained.

A desktop rule in the entitlement policy defines the users who are allowed per-session access to a machine from the rule's associated desktop group to run a full desktop session.

----- BrokerEntitlementPolicyRule Object

The BrokerEntitlementPolicyRule object represents a single desktop rule within the site's entitlement policy. It contains the following properties:

-- BrowserName (System.String)

Site-wide unique name identifying this desktop entitlement to other components (for example StoreFront).

-- ColorDepth (Citrix.Broker.Admin.SDK.ColorDepth?)

The color depth of any desktop session launched by the user from the entitlement. If null, the equivalent setting from the rule's desktop group is used.

-- Description (System.String)

Optional description of the rule. The text may be visible to the end user, for example, as a tooltip associated with the desktop entitlement.

-- DesktopGroupUid (System.Int32)

The unique ID of the desktop group to which the rule applies.

-- Enabled (System.Boolean)

Indicates whether the rule is enabled. A disabled rule is ignored when evaluating the site's entitlement policy.

-- ExcludedUserFilterEnabled (System.Boolean)

Indicates whether the excluded users filter is enabled. If the filter is disabled then any user entries in the filter are ignored when entitlement policy rules are evaluated.

-- ExcludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

The excluded users filter of the rule, that is, the users and groups who are explicitly denied an entitlement to a desktop session from this rule.

-- IconUid (System.Int32?)

The unique ID of the icon used to display the desktop entitlement to the user. If null, the equivalent setting from the rule's desktop group is used.

-- IncludedUserFilterEnabled (System.Boolean)

Indicates whether the included users filter is enabled. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to a desktop session by the rule.

-- IncludedUsers (Citrix.Broker.Admin.SDK.ChbUser[])

The included users filter of the rule, that is, the users and groups who are granted an entitlement to a desktop session by the rule.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

A collection of arbitrary key/value pairs that can be associated with the rule. The administrator can use these values for any purpose; they are not used by the site itself in any way.

-- Name (System.String)

The administrative name of the rule. Each rule in the site's entitlement policy must have a unique name (irrespective of whether they are desktop or application rules).

-- PublishedName (System.String)

The name of the desktop session entitlement as seen by the user. If null, the equivalent setting from the rule's desktop group is used.

-- SecureIcaRequired (System.Boolean?)

Indicates whether the rule requires the SecureICA protocol for desktop sessions launched using the entitlement. If null, the equivalent setting from the rule's desktop group is used.

-- SessionReconnection (Citrix.Broker.Admin.SDK.SessionReconnection)

Defines reconnection (roaming) behavior for sessions launched using this rule. Session reconnection control is an experimental and unsupported feature.

-- Uid (System.Int32)

The unique ID of the rule itself.

-- UUID (System.Guid)



UUID of the rule.

## Related topics

[New-BrokerEntitlementPolicyRule](#)

[Set-BrokerEntitlementPolicyRule](#)

[Rename-BrokerEntitlementPolicyRule](#)

[Remove-BrokerEntitlementPolicyRule](#)

## Parameters

**-Uid**<Int32>

Gets the desktop rule with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Name**<String>

Gets only desktop rules with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrowserName**<String>

Gets only desktop rules with browser names matching the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ColorDepth**<ColorDepth>

Gets only desktop rules with the specified color depth.

Valid values are \$null, FourBit, EightBit, SixteenBit, and TwentyFourBit.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Gets only desktop rules with the specified description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Gets only desktop rules that apply to the desktop group with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Gets only desktop rules that are in the specified state, either enabled (\$true), or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUser**<User>

Gets only desktop rules that have the specified user in their excluded users filter (whether the filter is enabled or not).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUserFilterEnabled**<Boolean>

Gets only desktop rules that have their excluded user filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconUid**<Int32>

Gets only desktop rules using the icon with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUser**<User>

Gets only desktop rules that have the specified user in their included users filter (whether the filter is enabled or not).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUserFilterEnabled**<Boolean>

Gets only desktop rules that have their included user filter enabled (\$true) or disabled (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName**<String>

Gets only desktop rules with the specified published name, that is, the desktop session entitlement name that the end user sees.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaRequired**<Boolean>

Gets only desktop rules that require the desktop session to use the SecureICA protocol (\$true) or not (\$false).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionReconnection**<SessionReconnection>

Gets only desktop rules with the specified session reconnection behavior.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UUID**<Guid>

Gets rules with the specified value of UUID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by - ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

None You cannot pipe input into this cmdlet.

### Return Values

Citrix.Broker.Admin.SDK.EntitlementPolicyRule

Get-BrokerEntitlementPolicyRule returns all desktop entitlement policy rules that match the specified selection criteria.

### Examples

#### ----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerEntitlementPolicyRule
```

Returns all desktop rules from the entitlement policy. This offers a complete description of the current site's entitlement policy with respect to desktops published from shared desktop groups.

#### ----- **EXAMPLE 2** -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Customer Support'
```

```
C:\PS> Get-BrokerEntitlementPolicyRule -DesktopGroupUid $dg.Uid
```

Returns all desktop rules in the entitlement policy that give users entitlements to desktop sessions in the Customer Support desktop group.

# Get-BrokerHostingPowerAction

Jan 04, 2017

Gets power actions queued for machines.

## Syntax

```
Get-BrokerHostingPowerAction [-Uid] <Int64> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerHostingPowerAction [[-MachineName] <String>] [-Action <PowerManagementAction>] [-ActionCompletionTime <DateTime>] [-ActionStartTime <DateTime>] [-ActualPriority <Int32>] [-BasePriority <Int32>] [-DNSName <String>] [-FailureReason <String>] [-HostedMachineName <String>] [-HypervisorConnectionName <String>] [-HypervisorConnectionUid <Int32>] [-Metadata <String>] [-RequestTime <DateTime>] [-State <PowerActionState>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Finds power actions matching the specified search criteria from the queue of all known power actions. These power actions can be waiting to be dealt with or can be part way through being processed by the relevant hypervisor, or they can be recently completed actions. Completed actions are removed from the queue after a configured period, the default being one hour.

If no search criteria are specified all actions in the queue are obtained.

A Hosting Power Action record defines the action that is to be performed or has been performed, the machine that the action is to be applied to, the priority of the action in relation to other actions in the queue, times for points in the life of the action, and any results if the action has completed.

For a detailed description of the queuing mechanism, see 'help about\_Broker\_PowerManagement'.

----- BrokerHostingPowerAction Object

The BrokerHostingPowerAction object represents an instance of a power action. It contains the following properties:

-- Action (Citrix.Broker.Admin.SDK.PowerManagementAction)

The power action to be performed. Possible values are: TurnOn, TurnOff, Shutdown, Reset, Restart, Suspend, Resume.

-- ActionCompletionTime (System.DateTime?)

The time when the power action was completed by the hypervisor connection.

-- ActionStartTime (System.DateTime?)

The time when the power action was started to be processed by the hypervisor.

-- ActualPriority (System.Int32)

The current priority of the operation after any priority boosting.

-- BasePriority (System.Int32)



The starting priority of the operation.

-- `DNSName` (System.String)

The fully qualified DNS name of the machine that the power action applies to.

-- `FailureReason` (System.String)

For failed power actions, an indication of the reason for the failure.

-- `HostedMachineName` (System.String)

The hypervisor's name for the machine that the power action applies to.

-- `HypervisorConnectionUid` (System.Int32)

The unique identifier of the hypervisor connection that is associated with the target machine.

-- `MachineName` (System.String)

The name of the machine that the power action applies to, in the form `domain\machine`.

-- `MetadataMap` (System.Collections.Generic.Dictionary<string, string>)

Metadata for this power action.

-- `RequestTime` (System.DateTime)

The timestamp of when the action was created and placed in the queue.

-- `State` (Citrix.Broker.Admin.SDK.PowerActionState)

The current state of this power action. Possible values are: Pending, Started, Completed, Failed, Canceled, Deleted, Lost.

-- `Uid` (System.Int64)

The unique identifier of the power action.

## Related topics

[New-BrokerHostingPowerAction](#)

[Set-BrokerHostingPowerAction](#)

[Remove-BrokerHostingPowerAction](#)

## Parameters

**-Uid**<Int64>

Gets only the single action record whose ID matches the specified value.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-MachineName**<String>

Gets only the records for actions that are for machines whose name (of the form domain\machine) matches the specified string.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Action**<PowerManagementAction>

Gets only action records with the specified action type.

Valid values are TurnOn, TurnOff, ShutDown, Reset, Restart, Suspend and Resume.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ActionCompletionTime**<DateTime>

Gets only action records reported as having completed successfully at the specified time. This is useful with advanced filtering; for more information, see [about\\_Broker\\_Filtering](#).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ActionStartTime**<DateTime>

Gets only action records reported as starting to be processed by the relevant hypervisor at the specified time. This is useful with advanced filtering; for more information, see [about\\_Broker\\_Filtering](#).

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ActualPriority<Int32>**

Gets only the records for actions whose current active priority matches the specified value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BasePriority<Int32>**

Gets only the records for actions whose original priority matches the specified value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DNSName<String>**

Gets only the records for actions that are for machines whose DNS name matches the specified string.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-FailureReason<String>**

Gets only the records for actions that have failed and whose failure reason string matches the specified string.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineName**<String>

Gets only the records for actions that are for machines whose Hosting Name (the machine name as understood by the hypervisor) matches the specified string.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionName**<String>

Gets only the records for actions for machines hosted via a hypervisor connection whose name matches the specified string.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionUid**<Int32>

Gets only the records for actions for machines hosted via a hypervisor connection whose ID matches the specified value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: `-Metadata "abc:x*"` matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RequestTime**<DateTime>

Gets only the records for actions created and added to the queue at the specified time. This is useful with advanced filtering; for more information, see [about\\_Broker\\_Filtering](#).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-State**<PowerActionState>

Gets only the records for actions with the specified current state.

Valid values are Pending, Started, Completed, Failed, Canceled, Deleted and Lost.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

None You cannot pipe input into this cmdlet.

### Return Values

Citrix.Broker.Admin.SDK.HostingPowerAction

Get-BrokerHostingPowerAction returns all power actions that match the specified selection criteria.

### Examples

#### ----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerHostingPowerAction
```

Fetches records for all known power actions either waiting to be processed, or currently being processed, or which have been processed in the last hour.

#### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerHostingPowerAction -State Pending -HypervisorConnectionName 'XenPool5'
```

Fetches records for all power actions that are waiting to be processed and where the action is for a virtual machine that is hosted by the hypervisor called 'XenPool5'.

# Get-BrokerHypervisorAlert

Jan 04, 2017

Gets hypervisor alerts recorded by the controller.

## Syntax

```
Get-BrokerHypervisorAlert -Uid <Int64> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerHypervisorAlert [-HostingServerName <String>] [-HypervisorConnectionUid <Int32>] [-Metadata <String>] [-Metric <AlertMetric>] [-Severity <AlertSeverity>] [-Time <DateTime>] [-TriggerInterval <TimeSpan>] [-TriggerLevel <Double>] [-TriggerPeriod <TimeSpan>] [-TriggerValue <Double>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Get-BrokerHypervisorAlert cmdlet gets alert objects reported by the hypervisors that the controller is monitoring.

Without parameters, Get-BrokerHypervisorAlert gets all of the alerts recorded. Use parameters to select which alerts are returned.

Once you have configured suitable alerts in your hypervisor, and configured the controller with your hypervisor details (see New-BrokerHypervisorConnection), the controller monitors each hypervisor for new alerts.

Four hypervisor alert metrics are recorded; these relate to the hypervisor host, not individual virtual machines:

- Cpu: Reports excessive CPU usage.
- Memory: Reports excessive memory usage.
- Network: Reports high network activity.
- Disk: Reports high disk activity.

Each alert also includes information about where and when the alert occurred, the severity of the alert (Red/Yellow), and the configuration of the triggered alert.

The following metrics are supported with these hypervisors:

- VMware ESX (Cpu, Memory, Network, Disk)
- Citrix XenServer (Cpu, Network)
- Microsoft Hyper-V (None)

----- BrokerHypervisorAlert Object

The BrokerHypervisorAlert represents a hypervisor alert object. It contains the following properties:

- HostingServerName (System.String)

The name of the hypervisor hosting this machine.



-- HypervisorConnectionUid (System.Int32)

The Uid of the hypervisor connection that reported this alert.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Metadata for this hypervisor alert.

-- Metric (Citrix.Broker.Admin.SDK.AlertMetric)

The metric this alert relates to: Cpu, Memory, Network or Disk.

-- Severity (Citrix.Broker.Admin.SDK.AlertSeverity)

Severity of the alert (Red or Yellow). Red is more serious than Yellow.

-- Time (System.DateTime)

Time that the alert occurred.

-- TriggerInterval (System.TimeSpan?)

Number of ticks (100ns) before the alert can be raised again.

-- TriggerLevel (System.Double?)

Threshold level that the alert was configured to trigger at.

-- TriggerPeriod (System.TimeSpan?)

Duration the value was above the trigger level.

-- TriggerValue (System.Double?)

The value of the monitored metric that triggered the alert.

-- Uid (System.Int64)

The unique internal identifier of this alert.

## Related topics

[New-BrokerHypervisorConnection](#)

## Parameters

**-Uid**<Int64>

Gets the hypervisor alert with the specified UID.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-HostingServerName**<String>

Gets alerts for the specified hosting hypervisor server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionUid**<Int32>

Gets alerts for the specified hypervisor connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metric**<AlertMetric>

Gets alerts for a specified metric.

Valid values are: Cpu, Memory, Network and Disk.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? |       |

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Severity**<AlertSeverity>

Gets alerts with the specified severity.

Valid values are: Red and Yellow.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Time**<DateTime>

Gets alerts that occurred at a specific time.

You can also use -Filter and relative comparisons; see the examples for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TriggerInterval**<TimeSpan>

Gets alerts with a specific trigger interval. This is the interval before the alert is raised again.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TriggerLevel**<Double>

Gets alerts with a specific trigger threshold level.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TriggerPeriod**<TimeSpan>

Gets alerts with a specific trigger period. This is the duration the threshold level was exceeded for, prior to the alert triggering.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TriggerValue**<Double>

Gets the value of the monitored metric that triggered the alert.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the

properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe objects to this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.HypervisorAlert

Get-BrokerHypervisorAlert returns an object for each matching alert.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerHypervisorAlert -HostingServerName TestHyp\* -Severity Red  
Returns all serious (Red) alerts for any hosting server with a name that starts with 'TestHyp'.

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerHypervisorAlert -Filter { Metric -eq 'Cpu' -and Time -ge '-1:0' }  
Returns all CPU usage alerts that occurred in the last hour.

# Get-BrokerHypervisorConnection

Jan 04, 2017

Gets hypervisor connections matching the specified criteria.

## Syntax

```
Get-BrokerHypervisorConnection [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerHypervisorConnection [[-Name] <String>] [-HypHypervisorConnectionUid <Guid>] [-IsReady <Boolean>] [-MachineCount <Int32>] [-MaxAbsoluteActiveActions <Int32>] [-MaxAbsoluteNewActionsPerMinute <Int32>] [-MaxAbsolutePvdPowerActions <Int32>] [-MaxPercentageActiveActions <Int32>] [-MaxPvdPowerActionsPercentageOfDesktops <Int32>] [-Metadata <String>] [-PreferredController <String>] [-State <HypervisorConnectionState>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Get-BrokerHypervisorConnection cmdlet gets hypervisor connections matching the specified criteria. If no parameters are specified this cmdlet enumerates all hypervisor connections.

----- BrokerHypervisorConnection Object

The BrokerHypervisorConnection represents hypervisor connection object. It contains the following properties:

-- Capabilities (System.String[])

The set of capabilities as reported by the hypervisor.

-- HypHypervisorConnectionUid (System.Guid)

The Guid that identifies the hypervisor connection.

-- IsReady (System.Boolean)

Indicates that the connection is ready to be used in the configuration of managed machines.

-- MachineCount (System.Int32)

Count of machines associated with this hypervisor connection.

-- MaxAbsoluteActiveActions (System.Int32?)

Maximum number of active power actions allowed at any one time (defined in the metadata named 'Citrix\_Broker\_MaxAbsoluteActiveActions' on the hypervisor connection in the Citrix Hosting Service).

-- MaxAbsoluteNewActionsPerMinute (System.Int32?)

Maximum number of new actions that can be fired off to the hypervisor in any one minute (defined in the metadata named 'Citrix\_Broker\_MaxAbsoluteNewActionsPerMinute' on the hypervisor connection in the Citrix Hosting Service).

-- MaxAbsolutePvdPowerActions (System.Int32?)

Maximum number of active Pvd power actions allowed at any one time (defined in the metadata named 'Citrix\_Broker\_MaxAbsolutePvdPowerActions' on the hypervisor connection in the Citrix Hosting Service).

-- MaxPercentageActiveActions (System.Int32?)

Maximum percentage of machines on the connection that can have active power actions at any one time (defined in the metadata named 'Citrix\_Broker\_MaxPowerActionsPercentageOfDesktops' on the hypervisor connection in the Citrix Hosting Service).

-- MaxPvdPowerActionsPercentageOfDesktops (System.Int32?)

Maximum percentage of machines on the connection that can be in personal VDisk image preparation mode at any one time (defined in the metadata named 'Citrix\_Broker\_MaxPvdPowerActionsPercentageOfDesktops' on the hypervisor connection in the Citrix Hosting Service).

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Collection of all the metadata associated to the hypervisor connection.

-- Name (System.String)

The display name of the hypervisor connection.

-- PreferredController (System.String)

The name of the controller which is preferred to be used, when available, to perform all communication to the hypervisor. The name is in DOMAIN\machine form. A preferred controller may have been automatically chosen when the hypervisor connection was created.

-- State (Citrix.Broker.Admin.SDK.HypervisorConnectionState)

The state of the hypervisor connection.

-- Uid (System.Int32)

Unique internal identifier of hypervisor connection.

## Related topics

[New-BrokerHypervisorConnection](#)

[Remove-BrokerHypervisorConnection](#)

[Set-BrokerHypervisorConnection](#)

## Parameters

**-Uid**<Int32>

Gets the hypervisor connection with the specified internal id.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |



|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Name**<String>

Gets hypervisor connections with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionUid**<Guid>

Gets hypervisor connections with the specified Guid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsReady**<Boolean>

Gets hypervisor connections with the specified value of the IsReady flag.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineCount**<Int32>

Gets hypervisor connections with the specified machine count.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-MaxAbsoluteActiveActions<Int32>**

Gets hypervisor connections with the specified MaxAbsoluteActiveActions value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxAbsoluteNewActionsPerMinute<Int32>**

Gets hypervisor connections with the specified MaxAbsoluteNewActionsPerMinute value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxAbsolutePvdPowerActions<Int32>**

Gets hypervisor connections with the specified MaxAbsolutePvdPowerActions value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxPercentageActiveActions<Int32>**

Gets hypervisor connections with the specified MaxPercentageActiveActions value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxPvdPowerActionsPercentageOfDesktops<Int32>**

Gets hypervisor connections with the specified MaxPvdPowerActionsPercentageOfDesktops value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata<String>**

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PreferredController<String>**

Gets hypervisor connections with the specified preferred controller. Specify the SAM name of the controller.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-State<HypervisorConnectionState>**

Gets hypervisor connections with the specified connection state. Values can be can be:

- o Unavailable - The broker is unable to contact the hypervisor.
- o InMaintenanceMode - The hosting server is in maintenance mode.
- o On - The broker is in contact with the hypervisor.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if

no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See `about_Broker_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None

## Return Values

Citrix.Broker.Admin.SDK.HypervisorConnection

Get-BrokerHypervisorConnection returns an object for each matching hypervisor connection.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS> $hvConn = Get-BrokerHypervisorConnection -Name "hvConnectionName"
```

Gets a hypervisor connection by name.

### ----- EXAMPLE 2 -----

```
c:\PS> $hvConn = Get-BrokerHypervisorConnection -PreferredController "domainName\controllerName"
```

Gets hypervisor connections by preferred controller.

### ----- EXAMPLE 3 -----

```
c:\PS> $machine = Get-BrokerMachine -Uid $machineUid
```

```
c:\PS> $hvConn = Get-BrokerHypervisorConnection -Uid $machine.HypervisorConnectionUid
```

Gets hypervisor connection used by a (power managed) machine.

# Get-BrokerIcon

Jan 04, 2017

Get stored icons.

## Syntax

```
Get-BrokerIcon -Uid <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerIcon [-Metadata <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Reads a specific icon by Uid, or enumerates icons by passing no Uid.

----- BrokerIcon Object

The BrokerIcon object represents a single instance of an icon. It contains the following properties:

-- EncodedIconData (System.String)

The Base64 encoded .ICO format of the icon.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Metadata for this command

-- Uid (System.Int32)

The UID of the icon itself.

## Related topics

[New-BrokerIcon](#)

[Remove-BrokerIcon](#)

## Parameters

**-Uid**<Int32>

Gets only the icon specified by unique identifier.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | 0     |



|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|           |       |

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

None Input cannot be piped to this cmdlet.

## Return Values

Citrix.Broker.Admin.SDK.Icon

Returns an Icon object for each enumerated icon.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerIcon  
Enumerate all icons.
```

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerIcon -Uid 1  
Get the icon with Uid 1.
```

# Get-BrokerImportedFTA

Jan 04, 2017

Gets the imported file type associations.

## Syntax

```
Get-BrokerImportedFTA [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerImportedFTA [[-ExtensionName] <String>] [-ContentType <String>] [-Description <String>] [-DesktopGroupUId <Int32>] [-Edit <String>] [-EditArguments <String>] [-EditExecutableName <String>] [-HandlerName <String>] [-Open <String>] [-OpenArguments <String>] [-OpenExecutableName <String>] [-PerceivedType <String>] [-Print <String>] [-PrintTo <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the file type associations the system imports from worker machines.

File type association associates a file extension (such as ".txt") with an application (such as Notepad). In a Citrix environment file type associations on a user device can be configured so that when an user clicks on a document it launches the appropriate published application. This is known as "content redirection".

Imported file type associations are different from configured file type associations. Imported file type associations are lists of known file type associations for a given desktop group. Configured file type associations are those that are actually associated with published applications for the purposes of content redirection.

Initially the system is not aware of any extensions, and they must be imported by the Citrix administrator. See the Update-BrokerImportedFTA cmdlet for more information.

After file type extensions are imported, this cmdlet lets the administrator review which file type associations the system is aware of. ImportedFTA objects are also used when configuring content redirection. See the New-BrokerConfiguredFTA cmdlet for more information.

The imported file type associations are grouped according to the desktop group to which they belong, because the system expects all machines in the same desktop group to have the same file type associations. That may not be true, however, across desktop groups.

Note that the ImportedFTA object has several fields that are not currently used. Only those fields that are shared with the ConfiguredFTA object are actually used in some capacity.

----- BrokerImportedFTA Object

The BrokerImportedFTA object represents a file type association imported from worker machines. It contains the following properties:

-- ContentType (System.String)

Content type of the file, such as "text/plain" or "application/vnd.ms-excel".

-- Description (System.String)

File type description, such as "Test Document", "Microsoft Word Text Document", etc.

-- DesktopGroupUid (System.Int32)

The desktop group this file type belongs to.

-- Edit (System.String)

Edit command with full path to executable: "C:\Program Files (x86)\Microsoft Office\Office12\WINWORD.EXE" /n /dde

-- EditArguments (System.String)

The arguments used for the 'edit' action on files of this type. These are extracted from the full edit command, and may be empty.

-- EditExecutableName (System.String)

The executable name extracted from the Edit property, no path included. This is used for matching with published apps' executable when searching for the list of extensions an application is capable of handling.

-- ExtensionName (System.String)

A single file extension, such as .txt. Unique within the scope of a desktop group.

-- HandlerName (System.String)

File type handler name, e.g. "Word.Document.8" or TXTFILE.

-- Open (System.String)

Open command with full path to executable: "C:\Program Files (x86)\Microsoft Office\Office12\WINWORD.EXE" /n /dde

-- OpenArguments (System.String)

The arguments used for the 'open' action on files of this type. These are extracted from the full open command, and may be empty.

-- OpenExecutableName (System.String)

The executable name extracted from the Open property, no path included. This is used for matching with published apps' executable when searching for the list of extensions an application is capable of handling.

-- PerceivedType (System.String)

Perceived type, such as "text".

-- Print (System.String)

Print command: "C:\Program Files (x86)\Microsoft Office\Office12\WINWORD.EXE" /x /n /dde

-- PrintTo (System.String)

PrintTo command: "C:\Program Files (x86)\Microsoft Office\Office12\WINWORD.EXE" /n /dde

-- Uid (System.Int32)

Unique internal identifier of imported file type association.

#### Related topics

[New-BrokerConfiguredFTA](#)

[Remove-BrokerImportedFTA](#)

[Update-BrokerImportedFTA](#)

#### Parameters

**-Uid**<Int32>

Gets only the imported file type associations with the specified unique identifier.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExtensionName**<String>

Gets only the imported file type associations with the specified extension name. For example, ".txt" or ".png".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ContentType**<String>

Gets only the imported file type associations with the specified content type (as listed in the Registry). For example, "application/msword".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Gets only the imported file type associations with the specified description (as listed in the Registry). For example, "Text Document" or "Microsoft Word text document".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid<Int32>**

Gets only the file type associations imported from a worker machine belonging to the specified desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Edit<String>**

Gets only the imported file type associations with the specified Edit command, that includes both the executable name and path, and any arguments to that executable.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-EditArguments<String>**

Gets only the imported file type associations with the specified arguments to the Edit command.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExecutableName<String>**

Gets only the imported file type associations with the specified executable for the Edit command.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HandlerName**<String>

Gets only the imported file type associations with the specified handler name (as listed in the Registry). For example, "TXT FILE" or "Word.Document.8".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Open**<String>

Gets only the imported file type associations with the specified Open command, that includes both the executable name and path, and any arguments to that executable.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OpenArguments**<String>

Gets only the imported file type associations with the specified arguments to the Open command.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OpenExecutableName**<String>

Gets only the imported file type associations with the specified executable for the Open command.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PerceivedType**<String>

Gets only the imported file type associations with the specified perceived type (as listed in the Registry). For example, "document".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Print**<String>

Gets only the imported file type associations with the specified Print command.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PrintTo**<String>

Gets only the imported file type associations with the specified PrintTo command.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>



When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See `about_Broker_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by `-ReturnTotalRecordCount`.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

### **-Filter**<String>

Gets records that match a PowerShell style filter expression. See `about_Broker_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

None This cmdlet does not accept any input from the pipeline.

### Return Values

Citrix.Broker.Admin.SDK.ImportedFTA

One or more ImportedFTA objects are returned as output.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerImportedFTA
```

Invoking this cmdlet with no arguments simply returns all of the imported file type association objects. By default, only the first 250 objects are returned. See the `-MaxRecordCount` and `-Skip` parameters for information about modifying this.

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerImportedFTA -ExtensionName ".txt"
```

Retrieves all imported file type associations that have the extension ".txt". Note that because imported file type associations are per-desktop group, multiple instances may be returned.

# Get-BrokerInstalledDbVersion

Jan 04, 2017

Gets a list of all available database schema versions for the Broker Service.

## Syntax

```
Get-BrokerInstalledDbVersion [-Upgrade] [-Downgrade] [-AdminAddress <String>][<CommonParameters>]
```

## Detailed Description

Gets the current version number of the Citrix Broker Service database schema when called with no parameters.

When called with the -Upgrade parameter, gets the service schema version numbers to which an upgrade could be performed.

When called with the -Downgrade parameter, gets the service schema version numbers to which a downgrade could be performed.

The SQL scripts to perform schema upgrades and downgrades can be obtained using the Get-BrokerDBVersionChangeScript cmdlet. Citrix recommends that where possible service schema upgrades are performed using Studio rather than manually via the SDK.

Only one of the -Upgrade or -Downgrade parameters may be supplied at once.

## Related topics

[Get-BrokerDBVersionChangeScript](#)

[Get-BrokerDBSchema](#)

## Parameters

**-Upgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be updated should be returned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Downgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be reverted should be returned.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

System.Version

Get-BrokerInstalledDBVersion returns database schema version numbers as requested.

**Notes**

If the command fails, the following errors can be returned.

**Error Codes**

-----

**InvalidParameterCombination**

Both the Upgrade and Downgrade flags were specified.

**NoOp**

The operation was successful but had no effect.

**NoDBConnections**

The database connection string for the Broker Service has not been specified.

**DatabaseError**

An error occurred in the service while attempting a database operation.

**DatabaseNotConfigured**

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

### Examples

#### ----- EXAMPLE 1 -----

```
C:\PS>Get-BrokerInstalledDBVersion
```

Gets the current Citrix Broker Service database schema version number.

#### ----- EXAMPLE 2 -----

```
C:\PS>Get-BrokerInstalledDBVersion -Upgrade
```

Get the versions of the Broker Service database schema for which upgrade scripts are supplied.

# Get-BrokerLease

Jan 04, 2017

Gets stored leases.

## Syntax

```
Get-BrokerLease [-Uid] <Int64> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerLease [[-Key] <String>] [-Expiration <DateTime>] [-LastModified <DateTime>] [-LeaseType <BrokerLeaseType>] [-OwnerSAMName <String>] [-OwnerSID <String>] [-OwnerUPN <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets leases filtered by specific Uid or Owner information.

----- BrokerLease Object

The BrokerLease object represents a single instance of a lease. It contains the following properties:

-- Expiration (System.DateTime)

The expiration time of the lease.

-- Key (System.String)

The SHA1 representation of the lease key.

-- LastModified (System.DateTime)

The modification time of the lease.

-- LeaseType (Citrix.Broker.Admin.SDK.BrokerLeaseType)

The type of lease.

-- OwnerSAMName (System.String)

The SAM name of the user associated with the lease.

-- OwnerSID (System.String)

The SID of the user associated with the lease.

-- OwnerUPN (System.String)

The UPN of the user associated with the lease.

-- Uid (System.Int64)

The UID of the lease itself.

-- Value (System.String)

The serialized lease data in XML.

## Related topics

[Remove-BrokerLease](#)

[Update-BrokerLocalLeaseCache](#)

## Parameters

**-Uid**<Int64>

Gets only the lease specified by unique identifier.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Key**<String>

Gets only the leases matching the specified lease key pattern.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Expiration**<DateTime>

Gets only the leases matching the specified expiration date and time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastModified**<DateTime>

Gets only the leases matching the specified modified date and time.



|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LeaseType**<BrokerLeaseType>

Gets only leases of a specific type. Possible values Enumeration, Launch.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OwnerSAMName**<String>

Gets only the leases associated with the specified Domain\User.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OwnerSID**<String>

Gets only the leases associated with the specified user SID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OwnerUPN**<String>

Gets only the leases associated with the specified user UPN.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if

no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See `about_Broker_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None Input cannot be piped to this cmdlet.

## Return Values

Citrix.Broker.Admin.SDK.Lease

Returns an Lease object for each enumerated lease.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $lease = Get-BrokerLease -Uid 1  
Gets the lease with internal Uid 1.
```

### ----- EXAMPLE 2 -----

```
C:\PS> $leases = Get-BrokerLease -OwnerSAMName Domain\User  
Gets the leases associated with the specified user.
```

# Get-BrokerMachine

Jan 04, 2017

Gets machines belonging to this site.

## Syntax

```
Get-BrokerMachine [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerMachine [-MachineName] <String> [-AgentVersion <String>] [-AllocationType <AllocationType>] [-ApplicationInUse <String>] [-AssignedClientName <String>] [-AssignedIPAddress <String>] [-AssociatedUserFullName <String>] [-AssociatedUserName <String>] [-AssociatedUserSID <String>] [-AssociatedUserUPN <String>] [-BrowserName <String>] [-CatalogName <String>] [-CatalogUid <Int32>] [-CatalogUUID <Guid>] [-ColorDepth <ColorDepth>] [-ControllerDNSName <String>] [-DeliveryType <DeliveryType>] [-Description <String>] [-DesktopCondition <String>] [-DesktopGroupName <String>] [-DesktopGroupUid <Int32>] [-DesktopGroupUUID <Guid>] [-DesktopKind <DesktopKind>] [-DesktopUid <Int32>] [-DNSName <String>] [-FaultState <MachineFaultState>] [-FunctionalLevel <FunctionalLevel>] [-HostedMachineId <String>] [-HostedMachineName <String>] [-HostingServerName <String>] [-HypervisorConnectionName <String>] [-HypervisorConnectionUid <Int32>] [-HypervisorConnectionGuid <Guid>] [-IconUid <Int32>] [-ImageOutOfDate <Boolean>] [-InMaintenanceMode <Boolean>] [-IPAddress <String>] [-IsAssigned <Boolean>] [-IsPhysical <Boolean>] [-LastConnectionFailure <ConnectionFailureReason>] [-LastConnectionTime <DateTime>] [-LastConnectionUser <String>] [-LastDeregistrationReason <DeregistrationReason>] [-LastDeregistrationTime <DateTime>] [-LastErrorReason <String>] [-LastErrorTime <DateTime>] [-LastHostingUpdateTime <DateTime>] [-LastPvdErrorReason <String>] [-LastPvdErrorTime <DateTime>] [-LoadIndex <Int32>] [-MachineInternalState <MachineInternalState>] [-Metadata <String>] [-OSType <String>] [-OSVersion <String>] [-PersistUserChanges <PersistUserChanges>] [-PowerActionPending <Boolean>] [-PowerState <PowerState>] [-ProvisioningType <ProvisioningType>] [-PublishedApplication <String>] [-PublishedName <String>] [-PvdEstimatedCompletionTime <DateTime>] [-PvdPercentDone <Int32>] [-PvdStage <PvdStage>] [-PvdUpdateStartTime <DateTime>] [-RegistrationState <RegistrationState>] [-ScheduledReboot <ScheduledReboot>] [-SecureIcaRequired <Boolean>] [-SessionAutonomouslyBrokered <Boolean>] [-SessionClientAddress <String>] [-SessionClientName <String>] [-SessionClientVersion <String>] [-SessionConnectedViaHostName <String>] [-SessionConnectedViaIP <String>] [-SessionCount <Int32>] [-SessionDeviceId <String>] [-SessionHardwareId <String>] [-SessionHidden <Boolean>] [-SessionKey <Guid>] [-SessionLaunchedViaHostName <String>] [-SessionLaunchedViaIP <String>] [-SessionProtocol <String>] [-SessionSecureIcaActive <Boolean>] [-SessionsEstablished <Int32>] [-SessionSmartAccessTag <String>] [-SessionsPending <Int32>] [-SessionStartTime <DateTime>] [-SessionState <SessionState>] [-SessionStateChangeTime <DateTime>] [-SessionSupport <SessionSupport>] [-SessionType <SessionType>] [-SessionUid <Int64>] [-SessionUserName <String>] [-SessionUserSID <String>] [-SID <String>] [-SummaryState <DesktopSummaryState>] [-SupportedPowerActions <String[]>] [-Tag <String>] [-UUID <Guid>] [-VMToolsState <VMToolsState>] [-WillShutdownAfterUse <Boolean>] [-WindowsConnectionSetting <WindowsConnectionSetting>] [-AssignedUserSID <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves machines matching the specified criteria. If no parameters are specified, this cmdlet enumerates all machines.

Get-BrokerMachine returns objects that combine machine configuration and state information.

For single-session machines, session information is displayed if present. If "fast user switching" is enabled, more than one session may be present on single-session machines. Because this cmdlet returns information only for a single session, if two

sessions are present it will return information about the brokered session (rather than, for example, an unbrokered direct RDP session). If there is no session running, session-related fields return \$null.

For multi-session machines, no session information about single sessions is displayed by this cmdlet, and so are always \$null. Get-BrokerSession can be used to get information about sessions on both multi-session and single-session machines.

To count machines, rather than retrieve full details of each machine, use Group-BrokerMachine instead.

See about\_Broker\_Filtering for information about advanced filtering options, and about\_Broker\_Machines for background information about machines.

#### ----- BrokerMachine Object

The machine object returned represents a physical or virtual machine, which has been configured in the site.

#### -- AgentVersion (System.String)

Version of the Citrix Virtual Delivery Agent (VDA) installed on the machine.

#### -- AllocationType (Citrix.Broker.Admin.SDK.AllocationType)

Describes how the machine is allocated to the user, can be Permanent or Random.

#### -- ApplicationsInUse (System.String[])

List of applications in use on the machine (in the form of browser name).

#### -- AssignedClientName (System.String)

The name of the endpoint client device that the machine has been assigned to.

#### -- AssignedIPAddress (System.String)

The IP address of the endpoint client device that the machine has been assigned to.

#### -- AssociatedUserFullNames (System.String[])

Full names of the users that have been associated with the machine (usually in the form "Firstname Lastname").

Associated users are the current user(s) for shared machines and the assigned users for private machines.

#### -- AssociatedUserNames (System.String[])

Usernames of the users that have been associated with the machine (in the form "domain\user").

Associated users are the current user(s) for shared machines and the assigned users for private machines.

#### -- AssociatedUserSIDs (System.String[])

The SIDs of the users that have been associated with the machine.

Associated users are the current user(s) for shared machines and the assigned users for private machines.

#### -- AssociatedUserUPNs (System.String[])

The User Principal Names of the users associated with the machine (in the form user@domain).

Associated users are the current user(s) for shared machines and the assigned users for private machines.

-- BrowserName (System.String)

Site-wide unique name identifying associated desktop to other components (for example StoreFront). This is typically non-null only for machines backing assigned private desktops.

-- Capabilities (System.String[])

List of the capabilities that the machine supports. Valid capabilities are:

- o MultiSession: Indicates an RDS- (Terminal Services-) based machine, which supports multiple active sessions from different users.

- o CBP1\_5: Indicates the machine uses the CBP 1.5 protocol for communication.

-- CatalogName (System.String)

Name of the catalog the machine is a member of.

-- CatalogUid (System.Int32)

UID of the catalog the machine is a member of.

-- CatalogUUID (System.Guid)

UUID of the catalog the machine is a member of.

-- ColorDepth (Citrix.Broker.Admin.SDK.ColorDepth?)

The color depth setting configured on the machine, possible values are:

\$null, FourBit, EightBit, SixteenBit, and TwentyFourBit.

-- ControllerDNSName (System.String)

The DNS host name of the controller that the machine is registered to.

-- DeliveryType (Citrix.Broker.Admin.SDK.DeliveryType?)

Denotes whether the machine delivers desktops only, apps only or both.

-- Description (System.String)

Description of the machine.

-- DesktopConditions (System.String[])

List of outstanding desktop conditions for the machine.

-- DesktopGroupName (System.String)

Name of the desktop group the machine is a member of.

-- DesktopGroupUid (System.Int32?)

UID of the desktop group the machine is a member of.

-- DesktopGroupUuiD (System.Guid?)

UuiD of the desktop group the machine is a member of.

-- DesktopKind (Citrix.Broker.Admin.SDK.DesktopKind?)

Deprecated.

Denotes whether the machine is private or shared. Use AllocationType instead.

-- DesktopUId (System.Int32?)

The UID of the associated desktop object.

-- DNSName (System.String)

The DNS host name of the machine.

-- FaultState (Citrix.Broker.Admin.SDK.MachineFaultState)

Summary state of any current fault state of the machine. Can be one of the following:

o None - No fault; machine is healthy.

o FailedToStart - Last power-on operation for machine failed.

o StuckOnBoot - Machine does not seem to have booted following power on.

o Unregistered - Machine has failed to register within expected period, or its registration has been rejected.

o MaxCapacity - Machine is reporting itself at maximum capacity.

-- FunctionalLevel (Citrix.Broker.Admin.SDK.FunctionalLevel?)

Functional level of the machine, if known.

-- HostedMachineId (System.String)

Unique ID within the hosting unit of the target managed machine.

-- HostedMachineName (System.String)

The friendly name of a hosted machine as used by its hypervisor. This is not necessarily the DNS name of the machine.

-- HostingServerName (System.String)

DNS name of the hypervisor that is hosting the machine if managed.

-- HypervisorConnectionName (System.String)

The name of the hypervisor connection that the machine has been assigned to, if managed.

-- HypervisorConnectionUId (System.Int32?)



The UID of the hypervisor connection that the machine's hosting server is accessed through.

-- HypHypervisorConnectionUid (System.Guid?)

The UUID of the hypervisor connection that the machine's hosting server is accessed through

-- IconUid (System.Int32?)

The UID of the machine's icon that is displayed in StoreFront.

-- ImageOutOfDate (System.Boolean?)

Denotes if the VM image for a hosted machine is out of date.

-- InMaintenanceMode (System.Boolean)

Denotes if the machine is in maintenance mode.

-- IPAddress (System.String)

The IP address of the machine.

-- IsAssigned (System.Boolean)

Denotes whether a private desktop has been assigned to a user/users, or a client name/address. Users can be assigned explicitly or by assigning on first use of the machine.

-- IsPhysical (System.Boolean)

This value is true if the machine is physical (ie not power managed by the Citrix Broker service, and false otherwise.

-- LastConnectionFailure (Citrix.Broker.Admin.SDK.ConnectionFailureReason)

The reason for the last failed connection between a client and the machine.

-- LastConnectionTime (System.DateTime?)

Time of the last detected connection attempt that either failed or succeeded.

-- LastConnectionUser (System.String)

The SAM name (in the form DOMAIN\user) of the user that last attempted a connection with the machine. If the SAM name is not available, the SID is used.

-- LastDeregistrationReason (Citrix.Broker.Admin.SDK.DeregistrationReason?)

The reason for the last deregistration of the machine with the broker. Possible values are:

AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

-- LastDeregistrationTime (System.DateTime?)

Time of the last deregistration of the machine from the controller.

-- LastErrorReason (System.String)

The reason for the last error detected in the machine.

-- LastErrorTime (System.DateTime?)

The time of the last detected error.

-- LastHostingUpdateTime (System.DateTime?)

Time of last update to any hosting data (such as power states) for this machine reported by the hypervisor connection.

-- LastPvdErrorReason (System.String)

The error text from the most recent failure of the Personal vDisk preparation process for this machine (if any).

-- LastPvdErrorTime (System.DateTime?)

The time of the most recent failure of the Personal vDisk preparation process for this machine (if any).

-- LoadIndex (System.Int32?)

Gives current effective load index for multi-session machines.

-- LoadIndexes (System.String[])

Gives the last reported individual load indexes that were used in the calculation of the LoadIndex value. Note that the LoadIndex value may have been subsequently adjusted due to session brokering operations. This value is only set for multi-session machines.

-- MachineInternalState (Citrix.Broker.Admin.SDK.MachineInternalState)

The internal state of the machine; reported while the machine is registered to a controller, plus some private Citrix Broker Service states while the machine is not registered.

-- MachineName (System.String)

DNS host name of the machine.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Any metadata that is associated with the machine.

-- OSType (System.String)

A string that can be used to identify the operating system that is running on the machine.

-- OSVersion (System.String)

A string that can be used to identify the version of the operating system running on the machine, if known.

-- PersistUserChanges (Citrix.Broker.Admin.SDK.PersistUserChanges)

Describes if and how user changes are persisted. Possible values are:

o OnLocal - Persist the user changes on the local disk of the machine.

o Discard - Discard user changes.

o OnPvd - Persist user changes on the Citrix Personal vDisk.

-- PowerActionPending (System.Boolean)

Indicates if there are any pending power actions for the machine.

-- PowerState (Citrix.Broker.Admin.SDK.PowerState)

The current power state of the machine. Possible values are: Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, resuming.

-- ProvisioningType (Citrix.Broker.Admin.SDK.ProvisioningType)

Describes how the machine was provisioned, possible values are:

o Manual: No automated provisioning.

o PVS: Machine provisioned by PVS (may be physical, blade, VM,...)

o MCS: Machine provisioned by MCS (machine must be VM)

-- PublishedApplications (System.String[])

List of applications published by the machine (displayed as browser names).

-- PublishedName (System.String)

The name of the machine that is displayed in StoreFront, if the machine has been published.

-- PvdEstimatedCompletionTime (System.DateTime?)

If preparation of the Personal vDisk is currently in progress for this machine, this reports an estimation of the time at which the process will be complete.

-- PvdPercentDone (System.Int32?)

If preparation of the Personal vDisk is currently in progress for this machine, this reports how far the process has got as a percentage. This value will be zero if preparation is not in progress.

-- PvdStage (Citrix.Broker.Admin.SDK.PvdStage)

For a machine supporting Personal vDisk technology (PvD), indicates the stage of the PvD image preparation.

-- PvdUpdateStartTime (System.DateTime?)

If preparation of the Personal vDisk is currently in progress for this machine, this reports when the update process began.

-- RegistrationState (Citrix.Broker.Admin.SDK.RegistrationState)

Indicates the registration state of the machine. Possible values are: Unregistered, Initializing, Registered, AgentError.

-- ScheduledReboot (Citrix.Broker.Admin.SDK.ScheduledReboot)

Indicates the state of any scheduled reboot operation for a machine. Possible values:

o None: No reboot is scheduled.

o Pending: Machine is awaiting reboot but is available for use.

o Draining: Machine is awaiting reboot and is unavailable for new sessions; reconnections to existing connections are still allowed, however.

o InProgress: Machine is actively undergoing a scheduled reboot. o Natural: Natural reboot in progress. Machine is awaiting a restart.

-- SecureIcaRequired (System.Boolean?)

Flag indicating whether SecureICA is required or not when starting a session on the machine.

-- SessionAutonomouslyBrokered (System.Boolean?)

Session property indicating if the current session is an HDX session established by direct connection without being brokered.

Session properties are always null for multi-session machines.

-- SessionClientAddress (System.String)

Session property indicating the IP address of the client connected to the machine.

Session properties are always null for multi-session machines.

-- SessionClientName (System.String)

Session property indicating the host name of the client connected to the machine.

Session properties are always null for multi-session machines.

-- SessionClientVersion (System.String)

Session property indicating the version of the Citrix Receiver on the connected client.

Session properties are always null for multi-session machines.

-- SessionConnectedViaHostName (System.String)

Session property indicating the host name of the connection gateway, router or client.

Session properties are always null for multi-session machines.

-- SessionConnectedViaIP (System.String)

Session property indicating the IP address of the connection gateway, router or client.

Session properties are always null for multi-session machines.

-- SessionCount (System.Int32)

Count of number of sessions on the machine.

-- SessionDeviceId (System.String)

Session property indicating a unique identifier for the client device that has most recently been associated with the current session.

Session properties are always null for multi-session machines.

-- SessionHardwareId (System.String)

Session property indicating a unique identifier for the client hardware that has been most recently associated with the current session.

Session properties are always null for multi-session machines.

-- SessionHidden (System.Boolean?)

Session property that indicates if a session is hidden.

Session properties are always null for multi-session machines.

-- SessionKey (System.Guid?)

Session property indicating the key of the current session.

Session properties are always null for multi-session machines.

-- SessionLaunchedViaHostName (System.String)

Session property that denotes the host name of the StoreFront server used to launch the current brokered session.

Session properties are always null for multi-session machines.

-- SessionLaunchedViaIP (System.String)

Session property that denotes the IP address of the StoreFront server used to launch the current brokered session.

Session properties are always null for multi-session machines.

-- SessionProtocol (System.String)

Session property that denotes the protocol that the current session is using, can be either HDX, RDP or Console. Console sessions on XenDesktop 5 VDAs appear with a blank protocol.

Session properties are always null for multi-session machines.

-- SessionSecureIcaActive (System.Boolean?)

Session property that indicates whether SecureICA is active on the current session or not.

Session properties are always null for multi-session machines.

-- SessionsEstablished (System.Int32)

Number of established sessions on this machine. For multi-session machines this excludes established sessions which have not yet completed their logon processing.

-- SessionSmartAccessTags (System.String[])

Session property that indicates the Smart Access tags for the current session.

Session properties are always null on multi-session machines.

-- SessionsPending (System.Int32)

Number of pending (brokered but not yet established) sessions on this machine. For multi-session machines this also includes established sessions which have not yet completed their logon processing.

-- SessionStartTime (System.DateTime?)

Session property that indicates the start time of the current session.

Session properties are always null on multi-session machines.

-- SessionState (Citrix.Broker.Admin.SDK.SessionState?)

Session property indicating the state of the current session, possible values are:

Other, PreparingSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession and Unknown. Session properties are always null for multi-session machines.

-- SessionStateChangeTime (System.DateTime?)

Session property indicating the time of the last state change of the current session.

Session properties are always null for multi-session machines.

-- SessionSupport (Citrix.Broker.Admin.SDK.SessionSupport)

Indicates the session support of the machine.

Possible values:

o SingleSession: Single-session only machine.

o MultiSession: Multi-session capable machine.

-- SessionType (Citrix.Broker.Admin.SDK.SessionType?)

Session property indicating the type of the current session.

Session properties are always null for multi-session machines.

-- SessionUid (System.Int64?)

Session property indicating the UID of the current session.

Session properties are always null for multi-session machines.

-- SessionUserName (System.String)

Session property indicates the name of the current session's user (in the form DOMAIN\user).

Session properties are always null for multi-session machines.

-- SessionUserSID (System.String)

Session property indicates the SID of the current session's user.

Session properties are always null for multi-session machines.

-- SID (System.String)

The SID of the machine.

-- SummaryState (Citrix.Broker.Admin.SDK.DesktopSummaryState)

Indicates the overall state of the desktop associated with the machine. The overall state is a result of other more specific states such as session state, registration state and power state. Possible values: Off, Unregistered, Available, Disconnected, InUse, Preparing.

-- SupportedPowerActions (System.String[])

A list of power actions supported by this machine.

-- Tags (System.String[])

A list of tags for the machine.

-- Uid (System.Int32)

UID of the machine object.

-- UUID (System.Guid)

UUID of the machine object.

-- VMToolsState (Citrix.Broker.Admin.SDK.VMToolsState)

State of the hypervisor tools present on the VM (if any).

Possible values are:

NotPresent, Unknown, NotStarted, Running.

-- WillShutdownAfterUse (System.Boolean)

Flag indicating if this machine is tainted and will be shut down after all sessions on the machine have ended. This flag is only ever true on power-managed, single-session machines.

Note: The machine will not shut down if it is in maintenance mode; it will shut down only after it is taken out of maintenance mode.

-- WindowsConnectionSetting (Citrix.Broker.Admin.SDK.WindowsConnectionSetting?)

The logon mode reported by Windows itself (multi-session machines only). For single-session machines the value is always

hardwired to LogonEnabled.

Possible values are:

LogonEnabled, Draining, DrainingUntilRestart and LogonDisabled.

### Related topics

[Group-BrokerMachine](#)

### Parameters

**-Uid**<Int32>

Gets a machine with a specific UID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName**<String>

Gets machines with a specific machine name (in the form domain\machine).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AgentVersion**<String>

Gets machines with a specific Citrix Virtual Delivery Agent version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AllocationType**<AllocationType>

Gets machines from catalogs with the specified allocation type.



|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationInUse<String>**

Gets machines running a specified published application (identified by browser name).

String comparisons are case-insensitive.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedClientName<String>**

Gets machines that have been assigned to the specific client name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedIPAddress<String>**

Gets machines that have been assigned to the specific IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserFullName<String>**

Gets machines with an associated user identified by their full name (usually 'first-name last-name').

Associated users are all current users of a desktop, plus the assigned users for private desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserName**<String>

Gets machines with an associated user identified by their user name (in the form 'domain\user').

Associated users are all current users of a desktop, plus the assigned users for private desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserSID**<String>

Gets machines with an associated user identified by their Windows SID.

Associated users are all current users of a desktop, plus the assigned users for private desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserUPN**<String>

Gets machines with an associated user identified by their User Principle Name (in the form 'user@domain').

Associated users are all current users of a desktop, plus the assigned users for private desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrowserName**<String>

Gets assigned machines backing desktop resources that have browser names matching the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CatalogName**<String>

Gets machines from the catalog with the specific name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CatalogUid**<Int32>

Gets machines from the catalog with the specific UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CatalogUUID**<Guid>

Gets machines from the catalog with the specific UUID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ColorDepth**<ColorDepth>

Gets machines configured with a specific color depth.

Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ControllerDNSName**<String>

Gets machines with a specific DNS name of the controller they are registered with.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DeliveryType**<DeliveryType>

Gets machines of a particular delivery type.

Valid values are AppsOnly, DesktopsOnly, DesktopsAndApps

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Gets machines with a specific description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopCondition**<String>

Gets machines with an outstanding desktop condition.

Valid values are:

- o CPU: Indicates the machine has high CPU usage
- o ICALatency: Indicates the network latency is high
- o UPMLogonTime: Indicates that the profile load time was high

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupName**<String>

Gets machines from a desktop group with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Gets machines from a desktop group with a specific UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUUID**<Guid>

Gets machines from a desktop group with a specific UUID.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopKind**<DesktopKind>

Deprecated: Use AllocationType parameter.

Gets machines of a particular kind.

Valid values are Private, Shared.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopUid**<Int32>

Gets the machine that corresponds to the desktop with the specific UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DNSName**<String>

Gets machines with the specific DNS name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-FaultState**<MachineFaultState>

Gets machines currently in the specified fault state.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-FunctionalLevel**<FunctionalLevel>

Gets machines with a specific FunctionalLevel.

Valid values are L5, L7, L7\_6

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineId**<String>

Gets machines with the specific machine ID known to the hypervisor.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineName**<String>

Gets machines with the specific machine name known to the hypervisor.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostingServerName**<String>

Gets machines by the name of the hosting hypervisor server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionName<String>**

Gets machines with the specific name of the hypervisor connection hosting them.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionUid<Int32>**

Gets machines with the specific UID of the hypervisor connection hosting them.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypHypervisorConnectionUid<Guid>**

Gets machines with the specific UUID of the hypervisor connection hosting them.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconUid<Int32>**

Gets machines by configured icon. Note that machines with a null IconUid use the icon of the desktop group.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ImageOutOfDate**<Boolean>

Gets machines depending on whether their disk image is out of date or not (for machines provisioned using MCS only).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InMaintenanceMode**<Boolean>

Gets machines by whether they are in maintenance mode or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IPAddress**<String>

Gets machines with a specific IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsAssigned**<Boolean>

Gets machines according to whether they are assigned or not. Machines may be assigned to one or more users or groups, a client IP address or a client endpoint name.

|           |       |
|-----------|-------|
| Required? | false |
|           |       |

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsPhysical**<Boolean>

Gets machines according to whether they can be power managed by XenDesktop or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastConnectionFailure**<ConnectionFailureReason>

Gets machines with a specific reason for the last recorded connection failure. This value is None if the last connection was successful or if there has been no attempt to connect to the desktop yet.

Valid values are None, SessionPreparation, RegistrationTimeout, ConnectionTimeout, Licensing, Ticketing, and Other.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastConnectionTime**<DateTime>

Gets machines on which a user session connection occurred at a specific time. This is the time at which the broker detected that the connection attempt either succeeded or failed.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastConnectionUser**<String>

Gets machines where a specific user name last attempted a connection (in the form 'domain\user').

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastDeregistrationReason**<DeregistrationReason>

Gets machines whose broker last recorded a specific deregistration reason.

Valid values are \$null, AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastDeregistrationTime**<DateTime>

Gets machines by the time that they were last deregistered.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastErrorReason**<String>

Gets machines with the specified last error reason.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

---

**-LastErrorTime**<DateTime>

Gets machines with the specified last error time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastHostingUpdateTime**<DateTime>

Gets machines with a specific time that the hosting information was last updated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastPvdErrorReason**<String>

Gets machines with the specified last Personal vDisk preparation error reason.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastPvdErrorTime**<DateTime>

Gets machines with the specified last Personal vDisk preparation error time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoadIndex**<Int32>

Gets machines by their current load index.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineInternalState**<MachineInternalState>

Gets machines with the specified internal state.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSType**<String>

Gets machines by the type of operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSVersion**<String>

Gets machines by the version of the operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PersistUserChanges**<PersistUserChanges>

Gets machines by the location where the user changes are persisted.

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the Pvd.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PowerActionPending**<Boolean>

Gets machines depending on whether a power action is pending or not.

Valid values are \$true or \$false.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PowerState**<PowerState>

Gets machines with a specific power state.

Valid values are Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ProvisioningType**<ProvisioningType>

Gets machines that are in a catalog with a particular provisioning type. Values can be:

- o Manual - No provisioning.
- o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).
- o MCS - Machine provisioned by MCS (machine must be VM).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedApplication**<String>

Gets machines with a specific application published to them (identified by its browser name).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName**<String>

Gets desktops with a specific published name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvdEstimatedCompletionTime**<DateTime>

If preparation of the Personal vDisk is currently in progress for this machine, this reports an estimation of the time at which the process will be complete.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvdPercent Done<Int32>**

Gets machines a specific percentage through the Personal vDisk preparation process.

This property is typically used with advanced filtering; see about\_Broker\_Filtering.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvdStage<PvdStage>**

Gets machines at a specific personal vDisk stage.

Valid values are None, Requested, Starting, Working and Failed.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvdUpdateStartTime<DateTime>**

If preparation of the Personal vDisk is currently in progress for this machine, this reports when the update process began.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |



---

**-RegistrationState**<RegistrationState>

Gets machines in a specific registration state.

Valid values are Unregistered, Initializing, Registered, and AgentError.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ScheduledReboot**<ScheduledReboot>

Gets machines according to their current status with respect to any scheduled reboots (for either scheduled desktop group reboots or image rollout purposes). Valid values are:

- o None - No reboot currently scheduled.
- o Pending - Reboot scheduled but machine still available for use.
- o Draining - Reboot scheduled. New logons are disabled, but reconnections to existing sessions are allowed.
- o InProgress - Machine is actively being rebooted.
- o Natural - Natural reboot in progress. Machine is awaiting a restart.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaRequired**<Boolean>

Gets machines configured with a particular SecureIcaRequired setting. Note that the machine setting of \$null indicates that the desktop group value is used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionAutonomouslyBrokered**<Boolean>

Gets machines according to whether their current session is autonomously brokered or not. Autonomously brokered sessions are HDX sessions established by direct connection without being brokered.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionClientAddress**<String>

Gets machines with a specific client IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionClientName**<String>

Gets machines with a specific client name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionClientVersion**<String>

Gets machines with a specific client version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

---

**-SessionConnectedViaHostName<String>**

Gets machines with a specific incoming connection host name. This is usually a proxy or Citrix Access Gateway server.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionConnectedViaIP<String>**

Gets machines with a specific incoming connection IP address.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionCount<Int32>**

Gets machines according to the total number of both pending and established user sessions on the machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionDeviceId<String>**

Gets machines with a specific client device ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-SessionHardwareId**<String>

Gets machines with a specific client hardware ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionHidden**<Boolean>

Gets machines depending on whether their sessions are hidden or not. Hidden sessions are treated as though they do not exist when launching sessions using XenDesktop; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionKey**<Guid>

Gets machine running the session with a specified unique key.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionLaunchedViaHostName**<String>

Gets machines with a specific host name of the StoreFront server from which the user launched the session.

Session properties are always null for multi-session machines.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionLaunchedViaIP**<String>

Gets machines with a specific IP address of the StoreFront server from which the user launched the session.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionProtocol**<String>

Gets machines with connections using a specific protocol, for example HDX, RDP, or Console.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionSecureIcaActive**<Boolean>

Gets machines depending on whether the current session uses SecureICA or not.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionsEstablished**<Int32>

Gets machines according to the number of established user sessions present on the machine.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionSmartAccessTag<String>**

Gets machines where the session has the specific SmartAccess tag.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionsPending<Int32>**

Get machines according to the number of pending user sessions for the machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionStartTime<DateTime>**

Gets machines with a specific session start time.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionState<SessionState>**

Gets machines with a specific session state.

Valid values are \$null, Other, PreparingSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession, and Unknown.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-SessionStateChangeTime**<DateTime>

Gets machines whose sessions last changed state at a specific time.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-SessionSupport**<SessionSupport>

Gets machines that have the specified session capability. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-SessionType**<SessionType>

Gets machines with a specific session state.

Session properties are always null for multi-session machines.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUid**<Int64>

Gets machines with a specific session UID (\$null for no session).

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUserName**<String>

Gets machines with a specific user name for the current session (in the form 'domain\user').

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUserSID**<String>

Gets machines with a specific SID of the current session user.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SID**<String>



Gets machines with a specific machine SID.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SummaryState**<DesktopSummaryState>

Gets machines with a specific summary state.

Valid values are Off, Unregistered, Available, Disconnected, and InUse.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SupportedPowerActions**<String[]>

A list of power actions supported by this machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Tag**<String>

Gets machines where the session has the given SmartAccess tag.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UUID**<Guid>

Gets machines with the specified value of UUID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-VMToolsState**<VMToolsState>

Gets machines with a specific VM tools state.

Valid values are NotPresent, Unknown, NotStarted, and Running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WillShutDownAfterUse**<Boolean>

Gets machines depending on whether they shut down after use or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WindowsConnectionSetting**<WindowsConnectionSetting>

Gets machines according to their current Windows connection setting (logon mode). Valid values are:

- o LogonEnabled - All logons are enabled.
- o Draining - New logons are disabled, but reconnections to existing sessions are allowed.
- o DrainingUntilRestart - Same as Draining, but setting reverts to LogonEnabled when machine next restarts.
- o LogonDisabled - All logons and reconnections are disabled.

This is a Windows setting and is not controlled by XenDesktop. It applies only to multi-session machines; for single-session

machines its value is always LogonEnabled.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedUserSID**<String>

Gets machines with the specific SID of the user to whom the desktop is assigned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by - ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

Citrix.Broker.Admin.SDK.Machine

Get-BrokerMachine returns an object for each matching desktop.

## Notes

It is generally better to compare dates and times using -Filter and relative comparisons. See about\_Broker\_Filtering and the examples in this topic for more information.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerMachine -PowerState Suspended
```

```
C:\PS> Get-BrokerMachine -Filter { PowerState -eq 'Suspended' }
```

These commands return all suspended machines. The second form uses advanced filtering (see about\_Broker\_Filtering).

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerMachine -DNSName '*.mydomain.mycompany.com'
```

This command returns all machines belonging to the DNS domain mydomain.mycompany.com.

### ----- EXAMPLE 3 -----

```
C:\PS> Get-BrokerMachine -Filter { RegistrationState -eq 'Registered' -and HypervisorConnectionUid -eq 5 }
```

This command returns all registered machines running on the specified hypervisor connection.

### ----- EXAMPLE 4 -----

```
C:\PS> Get-BrokerMachine -MachineName 'MyDomain\X*' | Remove-BrokerDesktopGroup -DesktopGroup 2
```

This command finds all of the machines in MyDomain with names beginning with X and removes them from the specified desktop group.

### ----- EXAMPLE 5 -----

```
C:\PS> Get-BrokerMachine -Filter { DesktopGroupUid -ne $null }
```

This command gets all desktops in a site. Use this instead of the deprecated Get-BrokerDesktop command.

# Get-BrokerMachineCommand

Jan 04, 2017

Get the list of commands queued for delivery to a desktop.

## Syntax

```
Get-BrokerMachineCommand -Uid <Int64> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerMachineCommand [-Category <String>] [-CommandName <String>] [-CompletionTime <DateTime>] [-MachineName <String>] [-MachineUid <Int32>] [-Metadata <String>] [-RequestTime <DateTime>] [-SendDeadline <TimeSpan>] [-SendDeadlineTime <DateTime>] [-SendTrigger <MachineCommandTrigger>] [-SessionUid <Int64>] [-State <MachineCommandState>] [-User <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Get the list of commands queued for delivery to a desktop. Commands are batched and can be configured to be delivered at various times during a desktop session's lifetime. Normally commands are sent within a few minutes of being queued, but it is also possible to queue a command for a user who is not currently logged on or a desktop that is currently switched off.

See `about_Broker_Filtering` for information about advanced filtering options.

----- BrokerMachineCommand Object

The command object returned represents a command handled by a specific service on a desktop as determined by the `Category` property.

-- `Category` (System.String)

Category of the command.

-- `CommandData` (System.Byte[])

Additional binary data included when the command is sent.

-- `CommandName` (System.String)

Name of the command.

-- `CompletionTime` (System.DateTime?)

Time at which the command was sent, expired or canceled.

-- `DesktopGroupNames` (System.String[])

List of desktop group names that the command was restricted to.

-- `MachineName` (System.String)

Name of the machine this command is targeted at.

-- `MachineUid` (System.Int32?)

Unique ID of the machine this command is targeted at.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Metadata for this command.

-- RequestTime (System.DateTime)

Time at which this command was created.

-- SendDeadline (System.TimeSpan)

Duration after which this command expires if it has not been sent yet.

-- SendDeadlineTime (System.DateTime?)

Time at which this command expires if it has not been sent yet.

-- SendTrigger (Citrix.Broker.Admin.SDK.MachineCommandTrigger?)

Event that triggers the sending of the command. Valid values are NextContact, Broker, LogOn, Logoff, Disconnect and Reconnect.

-- SessionUid (System.Int64?)

Unique ID of the session this command is targeted at.

-- State (Citrix.Broker.Admin.SDK.MachineCommandState)

Indicates whether the command is pending, sent, expired or canceled.

-- Synchronous (System.Boolean)

Flag that indicates if this is a synchronous command.

-- Uid (System.Int64)

Unique identifier of this machine command.

-- User (System.String)

Name of the user this command is targeted at.

## Related topics

[New-BrokerMachineCommand](#)

[Remove-BrokerMachineCommand](#)

## Parameters

**-Uid**<Int64>

Get only the command with the specified unique identifier.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Category**<String>

Get only commands targeted to the specified service category.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CommandName**<String>

Get only commands with the specified command name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CompletionTime**<DateTime>

Get only commands that entered the Sent, Failed, Canceled or Expired state at the specified time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName**<String>

Get only commands targeted to the specified machine.

|           |       |
|-----------|-------|
| Required? | false |
|           |       |

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineUid**<Int32>

Get only commands targeted to the specified machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RequestTime**<DateTime>

Get only commands that were requested at the specified time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SendDeadline**<TimeSpan>

Get only commands that expire after the specified time span.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SendDeadlineTime**<DateTime>

Get only commands that have the specified deadline time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SendTrigger**<MachineCommandTrigger>

Get only commands that are due to be sent when the specified trigger occurs. Valid values are NextContact, Broker, LogOn, Logoff, Disconnect and Reconnect.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUid**<Int64>

Get only commands targeted to the specified session.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-State**<MachineCommandState>

Get only commands in the specified state. Valid values are Pending, Sent, Failed, Canceled and Expired.

|           |       |
|-----------|-------|
| Required? | false |
|           |       |

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-User**<String>

Get only commands targeted to the specified user.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

None No parameter is accepted from the input pipeline.

### Return Values

Citrix.Broker.Admin.SDK.MachineCommand

Returns Command objects matching all specified selection criteria.

### Examples

----- **EXAMPLE 1** -----

Get-BrokerMachineCommand

Returns all pending, canceled, expired and sent commands.

----- **EXAMPLE 2** -----

Get-BrokerMachineCommand -State Pending

Returns all queued commands.

# Get-BrokerMachineConfiguration

Jan 04, 2017

Gets machine configurations defined for this site.

## Syntax

```
Get-BrokerMachineConfiguration [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerMachineConfiguration [[-Name] <String>] [-ConfigurationSlotUid <Int32>] [-LeafName <String>] [-Metadata <String>] [-DesktopGroupUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves machine configurations matching the specified criteria. If no parameters are specified this cmdlet enumerates all machine configurations.

Machine configurations contain binary arrays of settings data that are managed using SDK snap-ins. Each machine configuration is associated with a configuration slot and referenced by Name. The configuration slot restricts the settings that can be held by the machine configuration. For example, only configurations for Citrix User Profile Manager can be associated with the "User Profile Manager" slot.

See [about\\_Broker\\_Filtering](#) for information about advanced filtering options.

----- BrokerMachineConfiguration Object

The machine configuration object returned represents a named collection of related settings values that are applied to a desktop group.

-- ConfigurationSlotUid (System.Int32)

Uid of the associated configuration slot.

-- Description (System.String)

Optional description of the machine configuration.

-- DesktopGroupUids (System.Int32[])

List of desktop group Uids that this machine configuration has been added to.

-- LeafName (System.String)

Name of this machine configuration.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Map of metadata associated with this machine configuration.

-- Name (System.String)

Unique "SlotName\MachineConfigurationName" for this machine configuration.

-- Policy (System.Byte[])

A binary array of encoded settings.

-- Uid (System.Int32)

Uid of this machine configuration.

## Related topics

[New-BrokerMachineConfiguration](#)

[Set-BrokerMachineConfiguration](#)

[Rename-BrokerMachineConfiguration](#)

[Remove-BrokerMachineConfiguration](#)

[Add-BrokerMachineConfiguration](#)

## Parameters

**-Uid**<Int32>

Get only the machine configuration with the specified unique identifier.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Name**<String>

Get only the machine configuration with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ConfigurationSlotUid**<Int32>

Get only the machine configurations associated with the specified configuration slot.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? |       |



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LeafName**<String>

Get only the machine configurations that have the specified leaf name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Get only the machine configurations that have been assigned to the specified desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See `about_Broker_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.MachineConfiguration

Get-BrokerMachineConfiguration returns an object for each matching machine configuration.

**Examples**

----- **EXAMPLE 1** -----

Get-BrokerMachineConfiguration

Retrieves a list of every defined machine configuration.

----- **EXAMPLE 2** -----

```
Get-BrokerMachineConfiguration -Name Receiver\Engineering
```

Retrieves the machine configuration named "Receiver\Engineering".

----- **EXAMPLE 3** -----

```
Get-BrokerMachineConfiguration -Name UPM*
```

Retrieves a list of every machine configuration associated with the configuration slot named "UPM".

----- **EXAMPLE 4** -----

```
Get-BrokerMachineConfiguration -LeafName "Dept*"
```

Retrieves a list of every machine configuration with a LeafName that starts with "Dept", regardless of the associated configuration slot.

# Get-BrokerMachineStartMenuShortcutIcon

Jan 04, 2017

Retrieves a Start Menu Shortcut icon from the specified machine.

## Syntax

```
Get-BrokerMachineStartMenuShortcutIcon [-MachineName] <String> [-Path] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves the icon associated with a particular shortcut on a particular machine. This icon is usually used to help create a published application to access the shortcut.

## Related topics

[Get-BrokerMachine](#)

[New-BrokerIcon](#)

## Parameters

**-MachineName**<String>

Specify the name of the machine to use for icon retrieval for the specified shortcut path. The machine can be identified by DNS name, short name, SID, or name of the form domain\machine.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Path**<String>

The location of the shortcut in the specified machine whose icon is being fetched.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

---

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

System.String

Get-BrokerMachineStartMenuShortcutIcon generates a Base64 encoded string containing the icon for the specified shortcut. This can be used as input to New-BrokerIcon cmdlet.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $shortcuts = Get-BrokerMachineStartMenuShortcuts -MachineName 'MyDomain\MyMachine'  
C:\PS> $encodedIconData = Get-BrokerMachineStartMenuShortcutIcon -MachineName 'MyDomain\MyMachine' -Path $shortcuts[0].ShortcutPath  
C:\PS> $brokerIcon = New-BrokerIcon -EncodedIconData $encodedIconData  
C:\PS> Set-BrokerApplication 'Notepad' -IconUid $brokerIcon.Uid
```

This example retrieves all Start Menu Shortcuts from 'MyDomain\MyMachine', and then the icon for the first shortcut from the returned list. The icon is then associated with a published application called 'Notepad'.

# Get-BrokerMachineStartMenuShortcuts

Jan 04, 2017

Retrieves the Start Menu Shortcuts from the specified machine.

## Syntax

```
Get-BrokerMachineStartMenuShortcuts [-MachineName] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves the shortcuts defined for all the start menu items on a particular machine. The shortcuts obtained are from the 'All users' start menu; user-specific shortcuts are not found.

## Related topics

## Parameters

**-MachineName**<String>

Specify the name of the machine to use for shortcut retrieval. The machine can be identified by DNS name, short name, SID, or name of the form domain\machine.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

None or Citrix.Broker.Admin.SDK.StartMenuShortcut

Get-BrokerMachineStartMenuShortcuts generates an array of Citrix.Broker.Admin.SDK.StartMenuShortcut objects.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $shortcuts = Get-BrokerMachineStartMenuShortcuts -MachineName 'MyDomain\MyMachine'
```

This example retrieves all Start Menu Shortcuts from 'MyDomain\MyMachine'.



# Get-BrokerPowerTimeScheme

Jan 04, 2017

Gets power management time schemes for desktop groups.

## Syntax

```
Get-BrokerPowerTimeScheme [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerPowerTimeScheme [[-Name] <String>] [-DesktopGroupUid <Int32>] [-Metadata <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Finds power time schemes matching the specified search criteria. Each desktop group in the site can have a number of power time schemes associated with it, and these time schemes control how the power states of machines in the group are managed.

If no search criteria are specified all power time schemes for all desktop groups are obtained.

Each power time scheme covers one or more days of the week, and defines which hours of those days are considered peak times and which are off-peak times. In addition, the time scheme defines a pool size value for each hour of the day for the days of the week covered by the time scheme. No one desktop group can be associated with two or more time schemes that cover the same day of the week.

For any day of the week not covered by any power time scheme, it is assumed that all hours are off-peak and no pool size management is required for any of the hours.

For more information about the power policy mechanism and pool size management, see 'help about\_Broker\_PowerManagement'.

----- BrokerPowerTimeScheme Object

The BrokerPowerTimeScheme object represents a power time scheme, defining peak/off-peak hours and idle pool sizes for desktop groups. It contains the following properties:

-- DaysOfWeek (Citrix.Broker.Admin.SDK.TimeSchemeDays)

The days of the week for which this scheme applies to (Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Weekdays, Weekend).

-- DesktopGroupUid (System.Int32)

The desktop group that this time scheme is for.

-- DisplayName (System.String)

The name of this time scheme, as displayed in the Studio console.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Metadata for this power time scheme.

-- Name (System.String)

The unique name of this time scheme.

-- PeakHours (System.Boolean[])

A set of 24 boolean flag values, one for each hour of the day. The first value in the array relates to midnight to 00:59, the next one to 1 AM to 01:59 and so on, with the last array element relating to 11 PM to 11:59. If the flag is \$true it means that the associated hour of the day is considered a peak time; if it is \$false it means that it is considered off-peak.

-- PoolSize (System.Int32[])

A set of 24 integer values, one for each hour of the day. The first value in the array relates to midnight to 00:59, the next one to 1 AM to 01:59 and so on, with the last array element relating to 11 PM to 11:59. The value defines the number of machines (either as an absolute number or a percentage of the machines in the desktop group) that are to be maintained in a running state, whether they are in use or not. A value of -1 has special meaning: pool size management does not apply during such hours.

-- PoolUsingPercentage (System.Boolean?)

A boolean flag to indicate whether the integer values in the pool size array are to be treated as absolute values (if this value is \$false) or as percentages of the number of machines in the desktop group (if this value is \$true).

-- Uid (System.Int32)

Unique internal identifier of a time scheme.

## Related topics

[New-BrokerPowerTimeScheme](#)

[Set-BrokerPowerTimeScheme](#)

[Remove-BrokerPowerTimeScheme](#)

[Rename-BrokerPowerTimeScheme](#)

## Parameters

**-Uid**<Int32>

Gets only the power time scheme with the specified Uid.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Name**<String>

Gets only power time schemes with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Gets only the power time schemes associated with the specified desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: `-Metadata "abc:x"` matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See `about_Broker_Filtering` for details.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | False |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by - ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Property<String[]>**

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.PowerTimeScheme

Get-BrokerPowerTimeScheme returns all power time schemes that match the specified selection criteria.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerPowerTimeScheme  
Fetches all known power time schemes for all desktop groups in the site.

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerPowerTimeScheme -DesktopGroupId ( Get-BrokerDesktopGroup 'Sales Desktops' ).Uid  
Fetches all the power time schemes for the desktop group called 'Sales Desktops'.

# Get-BrokerPrivateDesktop

Jan 04, 2017

Get private desktops configured for this site.

## Syntax

```
Get-BrokerPrivateDesktop [-UId] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerPrivateDesktop [-MachineName] <String> [-AgentVersion <String>] [-AssignedClientName <String>] [-AssignedIPAddress <String>] [-ColorDepth <ColorDepth>] [-ControllerDNSName <String>] [-Description <String>] [-DesktopGroupUId <Int32>] [-DNSName <String>] [-HostedMachineId <String>] [-HostedMachineName <String>] [-HostingServerName <String>] [-HypervisorConnectionUId <Int32>] [-IconUId <Int32>] [-InMaintenanceMode <Boolean>] [-IPAddress <String>] [-IsAssigned <Boolean>] [-LastDeregistrationReason <DeregistrationReason>] [-LastDeregistrationTime <DateTime>] [-LastHostingUpdateTime <DateTime>] [-OSType <String>] [-OSVersion <String>] [-PowerState <PowerState>] [-PublishedName <String>] [-RegistrationState <RegistrationState>] [-SecureIcaRequired <Boolean>] [-SID <String>] [-Tag <String>] [-WillShutdownAfterUse <Boolean>] [-AssignedUserSID <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet is deprecated, please use the Get-BrokerMachine cmdlet instead.

Retrieve private desktops matching the specified criteria. If no parameters are specified, this cmdlet enumerates all private desktops.

Get-BrokerPrivateDesktop returns configuration information only for private desktops (a DesktopKind of 'Private'). For more state information about desktops, or other types of desktop, use the Get-BrokerMachine cmdlet instead.

For information about advanced filtering options, see [about\\_Broker\\_Filtering](#); for more information about desktops, see [about\\_Broker\\_Desktops](#).

----- BrokerPrivateDesktop Object

Private desktops are machines that have been configured with a DesktopKind of 'Private'. They are allocated to either a user/users or a client name/address (but cannot be allocated to both).

-- AgentVersion (System.String)

Version of the Citrix Virtual Delivery Agent (VDA) installed on the desktop.

-- AssignedClientName (System.String)

Client name the desktop has been assigned to.

-- AssignedIPAddress (System.String)

IP Address the desktop has been assigned to.

-- ColorDepth (Citrix.Broker.Admin.SDK.ColorDepth?)

The color depth setting configured on the desktop, possible values are:

\$null, FourBit, EightBit, SixteenBit, and TwentyFourBit.

-- ControllerDNSName (System.String)

The DNS host name of the controller that the desktop is registered to.

-- Description (System.String)

Description of the private desktop.

-- DesktopGroupUid (System.Int32)

Uid of the desktop group the desktop has been assigned to.

-- DNSName (System.String)

The DNS host name of the desktop.

-- HostedMachineId (System.String)

Unique ID within the hosting unit of the target managed desktop.

-- HostedMachineName (System.String)

The friendly name of a hosted desktop as used by its hypervisor. This is not necessarily the DNS name of the desktop.

-- HostingServerName (System.String)

DNS name of the hypervisor that is hosting the desktop if managed.

-- HypervisorConnectionUid (System.Int32?)

The UID of the hypervisor connection that the desktop has been assigned to, if managed.

-- IconUid (System.Int32?)

The UID of the desktop's icon that is displayed in StoreFront. If this is \$null then the desktop will use the icon specified by the desktop group.

-- InMaintenanceMode (System.Boolean)

Denotes whether the desktop is in maintenance mode.

-- IPAddress (System.String)

The IP address of the desktop.

-- IsAssigned (System.Boolean)

Denotes whether a private desktop has been assigned to a user/users, or a client name/address. Users can be assigned explicitly or by assigning on first use of the desktop.

-- LastDeregistrationReason (Citrix.Broker.Admin.SDK.DeregistrationReason?)

The reason for the last deregistration of the desktop with the broker. Possible values are:

AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

-- LastDeregistrationTime (System.DateTime?)

Time of the last deregistration of the desktop from the controller.

-- LastHostingUpdateTime (System.DateTime?)

Time of last update to any hosting data for this desktop reported by the hypervisor connection.

-- MachineName (System.String)

DNS host name of the machine associated with the desktop.

-- OSType (System.String)

A string that can be used to identify the operating system that is running on the desktop.

-- OSVersion (System.String)

A string that can be used to identify the version of the operating system running on the desktop, if known.

-- PowerState (Citrix.Broker.Admin.SDK.PowerState)

The current power state of the desktop. Possible values are: Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, Resuming.

-- PublishedName (System.String)

The name of the desktop that is displayed in StoreFront, if the desktop is published.

-- RegistrationState (Citrix.Broker.Admin.SDK.RegistrationState)

Indicates the registration state of the desktop. Possible values are: Unregistered, Initializing, Registered, AgentError.

-- SecureIcaRequired (System.Boolean?)

Flag indicating whether SecureICA is required or not when starting a session on the desktop.

-- SID (System.String)

Security identifier of the private desktop.

-- Uid (System.Int32)

Unique identifier of the private desktop.



-- WillShutdownAfterUse (System.Boolean)

Flag indicating whether this desktop is tainted and will be shutdown after all sessions on the desktop have ended. This flag should only ever be true on power managed, single-session desktops.

Note: The desktop will not shut down if it is in maintenance mode, but will shut down after the desktop is taken out of maintenance mode.

## Related topics

[Set-BrokerPrivateDesktop](#)

## Parameters

**-Uid**<Int32>

Gets desktops by Uid.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName**<String>

Gets desktops by machine name (in the form 'domain\machine').

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AgentVersion**<String>

Gets desktops with a specific Citrix Virtual Delivery Agent (VDA) version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedClient Name**<String>

Gets desktops assigned to a specific client name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedIPAddress**<String>

Gets desktops assigned to a specific IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ColorDepth**<ColorDepth>

Gets desktops configured with a specific color depth.

Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ControllerDNSName**<String>

Gets desktops by the DNS name of the controller they are registered with.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Gets desktops by description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid<Int32>**

Gets desktops from a desktop group with a specific Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DNSName<String>**

Gets desktops by DNS name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineId<String>**

Gets desktops by the machine id known to the hypervisor.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineName<String>**

Gets desktops by the machine name known to the hypervisor.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostingServerName**<String>

Gets desktops by the name of the hosting hypervisor server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionUid**<Int32>

Gets desktops by the uid of the hosting hypervisor connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconUid**<Int32>

Gets desktops by configured icon. Note that desktops with a \$null IconUid use the icon of the desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InMaintenanceMode**<Boolean>

Gets desktops by the InMaintenanceMode setting.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IPAddress**<String>

Get desktops by their IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsAssigned**<Boolean>

Gets desktops depending on whether they are assigned or not. Private desktops can be assigned to either a user/users or client names/addresses.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastDeregistrationReason**<DeregistrationReason>

Gets desktops whose broker last recorded a specific deregistration reason.

Valid values are \$null, AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastDeregistrationTime**<DateTime>

Gets desktops by the time that they were last deregistered.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastHostingUpdateTime**<DateTime>

Gets desktops by the time that the hosting information was last updated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSType**<String>

Gets desktops by the type of operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSVersion**<String>

Gets desktops by the version of the operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PowerState**<PowerState>

Gets desktops by power state.

Valid values are Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName**<String>

Gets desktops by published name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RegistrationState**<RegistrationState>

Gets desktops by registration state.

Valid values are Registered, Unregistered, and AgentError.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaRequired**<Boolean>

Gets desktops configured with a particular SecureIcaRequired setting. Note that the desktop setting of \$null indicates that the desktop group value is used.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-SID<String>**

Gets desktops by machine SID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Tag<String>**

Gets desktops tagged with the given tag.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WillShutDownAfterUse<Boolean>**

Gets desktops depending on whether they will be automatically shut down when the current session ends or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedUserSID<String>**

Gets desktops with the given assigned user (specified by SID).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|               |                                                         |
|---------------|---------------------------------------------------------|
| Required?     | false                                                   |
| Default Value | The default sort order is by name or unique identifier. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.PrivateDesktop

Get-BrokerPrivateDesktop returns an object for each matching private desktop.

## Notes

To compare dates/times, use `-Filter` and relative comparisons. See `about_Broker_Filtering` for details.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $list = 'Unmanaged','On','TurningOn','Resuming'
```

```
C:\PS> Get-BrokerPrivateDesktop -Filter { PowerState -in $list } | ft -a DNSName,PowerState
```

Get all private desktops that are turned on, or are turning on (assuming unmanaged desktops are powered on).

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerPrivateDesktop -Tag TestTag
```

Retrieve all private desktops tagged with the 'TestTag' tag.

# Get-BrokerRebootCycle

Jan 04, 2017

Gets one or more reboot cycles.

## Syntax

```
Get-BrokerRebootCycle -Uid <Int64> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerRebootCycle [-CatalogName <String>] [-CatalogUid <Int32>] [-DesktopGroupName <String>] [-DesktopGroupUid <Int32>] [-EndTime <DateTime>] [-MachinesCompleted <Int32>] [-MachinesFailed <Int32>] [-MachinesInProgress <Int32>] [-MachinesPending <Int32>] [-MachinesSkipped <Int32>] [-Metadata <String>] [-RebootDuration <Int32>] [-StartTime <DateTime>] [-State <RebootCycleState>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Get-BrokerRebootCycle cmdlet is used to enumerate reboot cycles that match all of the supplied criteria.

See about\_Broker\_Filtering for information about advanced filtering options.

### ----- BrokerRebootCycle Object

The reboot cycle object returned represents a single occurrence of the process of rebooting a portion (or all) of the machines in a desktop group.

-- CatalogName (System.String)

Name of the catalog whose machines are rebooted by this cycle if the cycle is associated with a catalog.

-- CatalogUid (System.Int32?)

Uid of the catalog whose machines are rebooted by this cycle if the cycle is associated with a catalog.

-- DesktopGroupName (System.String)

Name of the desktop group whose machines are rebooted by this cycle.

-- DesktopGroupUid (System.Int32)

Uid of the desktop group whose machines are rebooted by this cycle.

-- EndTime (System.DateTime?)

Time at which this cycle was completed, canceled or abandoned.

-- MachinesCompleted (System.Int32)

Number of machines successfully rebooted by this cycle.

-- MachinesFailed (System.Int32)

Number of machines issued with reboot requests where either the request failed or the operation did not complete within

the allowed time.

-- MachinesInProgress (System.Int32)

Number of machines issued with reboot requests but which have not yet completed the operation.

-- MachinesPending (System.Int32)

Number of outstanding machines to be rebooted during the cycle but on which processing has not yet started.

-- MachinesSkipped (System.Int32)

Number of machines scheduled for reboot during the cycle but which were not processed either because the cycle was canceled or abandoned or because the machine was unavailable for reboot processing throughout the cycle.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Map of metadata associated with this cycle.

-- RebootDuration (System.Int32)

Approximate maximum number of minutes over which the reboot cycle runs.

-- StartTime (System.DateTime)

Time of day at which this reboot cycle was started.

-- State (Citrix.Broker.Admin.SDK.RebootCycleState)

The execution state of this cycle.

-- Uid (System.Int64)

Unique ID of this reboot cycle.

-- WarningDuration (System.Int32)

Number of minutes to display the warning message for.

-- WarningMessage (System.String)

Warning message to display to users in active sessions prior to rebooting the machine.

-- WarningTitle (System.String)

Title of the warning message dialog.

Related topics

[Start-BrokerRebootCycle](#)

[Stop-BrokerRebootCycle](#)

Parameters

**-Uid**<Int64>

Gets reboot cycles that have the specified Uid.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CatalogName**<String>

Gets reboot cycles that relate to the named catalog.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CatalogUid**<Int32>

Gets reboot cycles that relate to the catalog with a particular Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupName**<String>

Gets reboot cycles that relate to the named desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Gets reboot cycles that relate to the desktop group with a particular Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-EndTime**<DateTime>

Gets reboot cycles that have the specified time at which the reboot cycle was completed, canceled or abandoned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachinesCompleted**<Int32>

Gets reboot cycles that have the specified count of machines successfully rebooted during the cycle.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachinesFailed**<Int32>

Gets reboot cycles that have the specified count of machines issued with reboot requests where either the request failed or the operation did not complete within the allowed time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachinesInProgress**<Int32>

Gets reboot cycles that have the specified count of machines issued with reboot requests but which have not yet completed the operation.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachinesPending**<Int32>

Gets reboot cycles that have the specified count of outstanding machines to be rebooted during the cycle but on which processing has not yet started.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachinesSkipped**<Int32>

Gets reboot cycles that have the specified count of machines scheduled for reboot during the cycle but which were not processed either because the cycle was canceled or abandoned or because the machine was unavailable for reboot processing throughout the cycle.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-RebootDuration**<Int32>

Gets reboot cycles that have the specified approximate maximum duration in minutes over which the reboot cycle runs.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartTime**<DateTime>

Gets reboot cycles that have the specified time at which the reboot cycle started.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-State**<RebootCycleState>

Gets reboot cycles that have the specified overall state of the reboot cycle. Valid values are Initializing, Active, Completed, Canceled, and Abandoned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None Input cannot be piped to this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.RebootCycle

Returns matching reboot cycles.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerRebootCycle  
Enumerate all reboot cycles.

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerRebootCycle -State Completed  
Enumerates all reboot cycles that have successfully completed.

----- **EXAMPLE 3** -----

```
C:\PS> Get-BrokerRebootCycle -DesktopGroupName CallCenter
```

Enumerates all reboot cycles related to the desktop group named CallCenter.

# Get-BrokerRebootSchedule

Jan 04, 2017

Gets one or more reboot schedules.

## Syntax

```
Get-BrokerRebootSchedule [-DesktopGroupUid] <Int32> [-Property <String[]>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Get-BrokerRebootSchedule [[-DesktopGroupName] <String>] [-Active <Boolean>] [-Day <RebootScheduleDays>] [-Enabled  
<Boolean>] [-Frequency <RebootScheduleFrequency>] [-RebootDuration <Int32>] [-StartTime <TimeSpan>] [-  
ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>]  
[-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Get-BrokerRebootSchedule cmdlet is used to enumerate desktop group reboot schedules that match all of the supplied criteria.

A reboot schedule can be configured to cause all of the machines in a desktop group to be rebooted at a particular time each day or each week, with the reboot of the individual machines spread out over the duration of the whole reboot cycle. A specific warning message can be configured to be displayed to users who are running sessions on the machines being rebooted. Note that each desktop group can only have a single reboot schedule configured.

See about\_Broker\_Filtering for information about advanced filtering options.

----- BrokerRebootSchedule Object

The reboot schedule object returned represents a regularly scheduled reboot of machines in a desktop group.

-- Active (System.Boolean)

True if there is an active reboot cycle for this schedule, false otherwise.

-- Day (Citrix.Broker.Admin.SDK.RebootScheduleDays)

When the frequency is weekly, day of the week on which the schedule reboot starts.

-- DesktopGroupName (System.String)

Name of the desktop group rebooted by this schedule.

-- DesktopGroupUid (System.Int32)

Uid of the desktop group rebooted by this schedule.

-- Enabled (System.Boolean)

True if this schedule is currently enabled, false otherwise.

-- Frequency (Citrix.Broker.Admin.SDK.RebootScheduleFrequency)

Whether the schedule runs daily or weekly.

-- RebootDuration (System.Int32)

Approximate maximum number of minutes over which the scheduled reboot cycle runs.

-- StartTime (System.TimeSpan)

Time of day at which the scheduled reboot cycle starts.

-- WarningDuration (System.Int32)

Number of minutes to display the warning message for.

-- WarningMessage (System.String)

Warning message to display to users in active sessions prior to rebooting the machine.

-- WarningTitle (System.String)

Title of the warning message dialog.

Related topics

[Set-BrokerRebootSchedule](#)

[New-BrokerRebootSchedule](#)

[Remove-BrokerRebootSchedule](#)

[Get-BrokerRebootCycle](#)

Parameters

**-DesktopGroupUid**<Int32>

Gets the reboot schedule for the desktop group having this Uid.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupName**<String>

Gets the reboot schedule for the desktop group having this name.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Active**<Boolean>

Gets desktop group reboot schedules according to whether they are currently active or not. A schedule is active if there is a reboot cycle currently running that was started as a result of the schedule.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Day**<RebootScheduleDays>

Gets the reboot schedules set to run on the specified day (one of Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Gets the reboot schedules with the specified Enabled value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Frequency**<RebootScheduleFrequency>

Gets the reboot schedules with the specified frequency (either Weekly or Daily).

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-RebootDuration**<Int32>

Gets the reboot schedules with the specified duration.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartTime**<TimeSpan>

Gets the reboot schedules with the specified start time (HH:MM).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | 250   |



|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by - ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None Input cannot be piped to this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.RebootSchedule

Returns matching reboot schedules.

Examples

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerRebootSchedule  
Enumerates all of the reboot schedules.

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerRebootSchedule -Enabled \$false -Frequency Daily  
Enumerates all disabled reboot schedules that are scheduled to run daily.

----- **EXAMPLE 3** -----

C:\PS> Get-BrokerRebootSchedule -DesktopGroupUid 11  
Returns the unique reboot schedule for the desktop group having the Uid 11.

# Get-BrokerRemotePCAccount

Jan 04, 2017

Get RemotePCAccount entries configured for this site.

## Syntax

```
Get-BrokerRemotePCAccount -Uid <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerRemotePCAccount [-AllowSubfolderMatches <Boolean>] [-CatalogUid <Int32>] [-OU <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves RemotePCAccounts matching the specified criteria. If no parameters are specified this cmdlet enumerates all RemotePCAccounts. Each RemotePCAccount object defines a set of machines either by machine name patterns or by where the machines are placed in Active Directory, and which RemotePC catalog the machines are to be associated with when they are discovered.

----- BrokerRemotePCAccount Object

RemotePCAccounts define a set of machines either by machine name patterns or by where the machines are placed in Active Directory, and which RemotePC catalog the machines are to be associated with when they are discovered.

-- AllowSubfolderMatches (System.Boolean)

Specifies whether machines subfolders of specified AD OUs are to be considered part of the RemotePCAccount.

-- CatalogUid (System.Int32)

The Uid of the RemotePC catalog to which machines in the RemotePCAccount automatically join during registration.

-- MachinesExcluded (System.String[])

A list of machines which are to be excluded from the RemotePCAccount. Wildcard matching is supported.

-- MachinesIncluded (System.String[])

A list of machines which are to be included in the RemotePCAccount. Wildcard matching is supported.

-- OU (System.String)

Machines within this specified AD OU are considered part of the RemotePCAccount, unless they are in they match the MachinesExcluded

-- Uid (System.Int32)

The Uid of the RemotePCAccount object.

## Related topics

[New-BrokerRemotePCAccount](#)

## Set-BrokerRemotePCAccount

## Remove-BrokerRemotePCAccount

### Parameters

#### **-Uid**<Int32>

Gets the RemotePCAccount with the specified unique ID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-AllowSubfolderMatches**<Boolean>

Gets RemotePCAccounts with the specified value of AllowSubfolderMatches.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-CatalogUid**<Int32>

Gets RemotePCAccounts belonging to the specified Remote PC catalog.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-OU**<String>

Gets the RemotePCAccount with the specified OU.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-Sort By**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.RemotePCAccount

Get-BrokerRemotePCAccount returns an object for each matching RemotePCAccount.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerRemotePCAccount  
Find all RemotePCAccounts.
```

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerRemotePCAccount -CatalogUid 42  
Find RemotePCAccounts belonging to Remote PC catalog 42.
```

# Get-BrokerResource

Jan 04, 2017

Gets resources that a user can broker connections to.

## Syntax

```
Get-BrokerResource [-User] <String> [-Groups <String[]>] [-ClientName <String>] [-ClientIP <String>] [-ViaAG <Boolean>] [-SmartAccessTags <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieve a list of resources that a user has access to, taking into account the site access policy, configuration of desktop groups, assignments, entitlements, and applications.

What a user has access to depends on a number of attributes:

- User's name or security identifier.
- Groups that the user is a member of (names or security identifiers).
- IP address of the client the user connects from.
- Name of the client that the user connects from.
- Whether the user is connecting via Citrix Access Gateway.
- SmartAccess tags when connecting via Citrix Access Gateway.

You must always specify the user's name or security identifier, but you will not always be able to predict what some of the other values will be. By omitting these values the corresponding access checks are ignored.

Consider for example, a site configuration that uses IP address ranges to allow access to private desktop A when connecting from the local network and private desktop B when connecting from home. Running this cmdlet without specifying a client IP address would return both A and B.

The output of this cmdlet depends on the available resources:

- Assigned private desktops are returned as PrivateDesktop objects.
- Shared desktops are returned as EntitlementPolicyRule objects.
- Assign-On-First-Use desktops that have not been assigned yet are returned as AssignmentPolicyRule objects.
- Application resources produce Application objects.

If more than one type of resource is available, the output pipeline contains a mixture of the above objects, in no particular order.

Only resources accessible based on the specified parameters, and visible to the administrator running this cmdlet are returned.

## Related topics



## Parameters

### **-User**<String>

Gets resources given the specified user name or security identifier.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-Groups**<String[]>

Get resources accessible given a list of group names or security identifiers.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-Client Name**<String>

Get resources given the specified client name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-Client IP**<String>

Get resources given the specified client IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ViaAG**<Boolean>

Gets resources given the specified ViaAG setting.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SmartAccessTags**<String[]>

Get resources given the specified SmartAccess tags.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.PrivateDesktop

Get-BrokerResource returns a PrivateDesktop for each accessible assigned private desktop.Citrix.Broker.Admin.SDK.EntitlementPolicyRule

Get-BrokerResource returns an EntitlementPolicyRule object for each accessible entitlement to a shared desktop.Citrix.Broker.Admin.SDK.AssignmentPolicyRule

Get-BrokerResource returns an AssignmentPolicyRule object for each accessible Assign-On-First-Use desktop.Citrix.Broker.Admin.SDK.Application

Get-BrokerResource returns an Application object for each accessible application.

## Examples

### ----- EXAMPLE 1 -----

Get-BrokerResource -User MYDOMAIN\User1 -Group MYDOMAIN\Accounts,MYDOMAIN\Managers  
List resources visible by User1 assuming membership of a couple of groups.

### ----- EXAMPLE 2 -----

```
[int[]]$groups = (Get-BrokerResource -User MYDOMAIN\User1 | %{ $_.DesktopGroupUid })  
Get-BrokerDesktopGroup -Filter { Uid -in $groups } -Property Uid,Name  
Get all of the desktop groups supporting the resources accessible by User1, outputting the uid and name of each desktop group.
```

# Get-BrokerScopedObject

Jan 04, 2017

Gets the details of the scoped objects for the Broker Service.

## Syntax

```
Get-BrokerScopedObject -ScopeId <Guid> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerScopedObject [-Description <String>] [-ObjectId <String>] [-ObjectName <String>] [-ObjectType  
<ScopedObjectType>] [-ScopeName <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-  
SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns a list of directly scoped objects including the names and identifiers of both the scope and object as well as the object description for display purposes.

There will be at least one result for every directly scoped object. When an object is associated with multiple scopes the output contains one result per scope duplicating the object details.

No records are returned for the All scope, though if an object is not in any scope a result with a null ScopeId and ScopeName will be returned.

----- BrokerScopedObject Object

A scoped, or scopeable object configured in the Broker.

-- Description (System.String)

Description of the object (possibly \$null if the object type does not have a description).

-- ObjectId (System.String)

Unique identifier of the object.

-- ObjectName (System.String)

Display name of the object.

-- ObjectType (Citrix.Broker.Admin.SDK.ScopedObjectType)

Type of the object this entry relates to.

-- ScopeId (System.Guid?)

Specifies the unique identifier of the scope.

-- ScopeName (System.String)

Specifies the display name of the scope.

## Related topics

## Parameters

### **-ScopeId**<Guid>

Gets scoped object entries for the given scope identifier.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-Description**<String>

Gets scoped object entries for objects with the specified description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-ObjectId**<String>

Gets scoped object entries for objects with the specified object identifier.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-ObjectName**<String>

Gets scoped object entries for objects with the specified object identifier.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ObjectType**<ScopedObjectType>

Gets scoped object entries for objects of the given type.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ScopeName**<String>

Gets scoped object entries with the given scope name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Return Values

Citrix.Broker.Sdk.ScopedObject

The Get-BrokerScopedObject command returns an object containing the following properties:

ScopeId <Guid?>

Specifies the unique identifier of the scope.

ScopeName <String>

Specifies the display name of the scope.

ObjectType <ScopedObjectType>

Type of the object this entry relates to.

ObjectId <String>

Unique identifier of the object.

ObjectName <String>

Display name of the object

Description <String>

Description of the object (possibly \$null if the object type does not have a description).

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

UnknownObject



One of the specified objects was not found.

PartialData

Only a subset of the available data was returned.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

CouldNotQueryDatabase

The query required to get the database was not defined.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-BrokerScopedObject -ObjectType Scheme
```

```
ScopeId : eff6f464-f1ee-4442-add3-99982e0cec01
ScopeName : Sales
ObjectType : Scheme
ObjectId : cd4174ee-9e4b-4e57-b126-9dbf757fe493
ObjectName : MyExampleScheme
Description : Test scheme
```

ScopeId : 304e0fa7-d390-47f0-a94f-7e956a324c41  
ScopeName : Finance  
ObjectType : Scheme  
ObjectId : cd4174ee-9e4b-4e57-b126-9dbf757fe493  
ObjectName : MyExampleScheme  
Description : Test scheme

ScopeId :  
ScopeName :  
ObjectType : Scheme  
ObjectId : 5062e46b-71bc-4ac9-901a-30fe6797e2f6  
ObjectName : AnotherScheme  
Description : Another scheme in no scopes

Gets all of the scoped objects with type Scheme. The example output shows a scheme object (MyExampleScheme) in two scopes Sales and Finance, and another scheme (AnotherScheme) that is not in any scope. The ScopeId and ScopeName values returned are null in the final record.

# Get-BrokerServiceAddedCapability

Jan 04, 2017

Gets any added capabilities for the Broker Service on the controller.

## Syntax

```
Get-BrokerServiceAddedCapability [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables updates to the Broker Service on the controller to be detected.

There is no requirement for a database connection to be configured in order for this command to be used.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

None You cannot pipe input into this cmdlet.

### Return Values

System.String

String containing added capabilities.

### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

##### ----- **EXAMPLE 1** -----

```
C:\PS>Get-BrokerServiceAddedCapability
```

Get the added capabilities of the Broker Service.

# Get-BrokerServiceInstance

Jan 04, 2017

Gets the service instance entries for the Broker Service.

## Syntax

```
Get-BrokerServiceInstance [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns service interfaces published by instances of the Broker Service. Each instance of a service publishes multiple interfaces with distinct interface types, and each of these interfaces is represented as a ServiceInstance object. Service instances can be used to register the service with a central configuration service so that other services can use the functionality.

You do not need to configure a database connection to use this command.

## Related topics

[Get-BrokerServiceStatus](#)

[Reset-BrokerServiceGroupMembership](#)

## Parameters

**-AdminAddress** <String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Return Values

Citrix.Broker.Sdk.ServiceInstance

The Get-BrokerServiceInstance command returns an object containing the following properties.

ServiceGroupUid <Guid>

Specifies the unique identifier for the service group of which the service is a member.

ServiceGroupName <String>

Specifies the name of the service group of which the service is a member.

#### ServiceInstanceUID <Guid>

Specifies the unique identifier for registered service instances, which are service instances held by and obtained from a central configuration service. Unregistered service instances do not have unique identifiers.

#### ServiceType <String>

Specifies the service instance type. For this service, the service instance type is always Broker.

#### Address

Specifies the address of the service instance. The address can be used to access the service and, when registered in the central configuration service, can be used by other services to access the service.

#### Binding

Specifies the binding type that must be used to communicate with the service instance. In this release of XenDesktop, the binding type is always 'wcf\_HTTP\_kerb'. This indicates that the service provides a Windows Communication Foundation endpoint that uses HTTP binding with integrated authentication.

#### Version

Specifies the version of the service instance. The version number is used to ensure that the correct versions of the services are used for communications.

#### ServiceAccount <String>

Specifies the Active Directory account name for the machine on which the service instance is running. The account name is used to provide information about the permissions required for interservice communications.

#### ServiceAccountSid <String>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

#### InterfaceType <String>

Specifies the interface type. Each service can provide multiple service instances, each for a different purpose, and the interface defines the purpose. Available interfaces are:

SDK - for PowerShell operations

InterService - for operations between different services

Peer - for communications between services of the same type

#### Metadata <Citrix.Broker.Sdk.Metadata[]>

The collection of metadata associated with registered service instances, which are service instances held by and obtained from a central configuration service. Metadata is not stored for unregistered service instances.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

##### ----- **EXAMPLE 1** -----

```
c:\PS>Get-BrokerServiceInstance
```

Get all instances of the Broker Service running on the specified machine. For remote services, use the AdminAddress parameter to define the service for which the interfaces are required. If the AdminAddress parameter has not been specified for the runspace, service instances running on the local machine are returned.

# Get-BrokerServiceStatus

Jan 04, 2017

Gets the current state of the Broker Service on the controller.

## Syntax

```
Get-BrokerServiceStatus [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables the status of the Broker Service on the controller to be determined. If the service has multiple data stores it will return the overall state as an aggregate of all the data store states. For example, if the site data store status is OK and the secondary data store status is DBUnconfigured then it will return DBUnconfigured. Before using this command, you don't have to configure the database connection to the Service.

## Related topics

[Set-BrokerDBConnection](#)

[Test-BrokerDBConnection](#)

[Get-BrokerDBConnection](#)

[Get-BrokerDBSchema](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Get-BrokerServiceStatus command returns an object containing the status of the Broker Service together with extra diagnostics information.



#### DBUnconfigured

The Broker Service does not have a database connection configured.

#### DBRejectedConnection

The database rejected the logon attempt from the Broker Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

#### InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Broker Service schema has not been added to the database.

#### DBNotFound

The specified database could not be located with the configured connection string.

#### DBMissingOptionalFeature

The Broker is connected to a database that is valid, but it does not have the full functionality required for optimal performance. Upgrading the database is advisable.

#### DBMissingMandatoryFeature

The Broker is connected to a database that is valid, but it does not have the full functionality required so the Broker cannot function. Upgrading the database is required.

#### DBNewerVersionThanService

The version of the Broker Service currently in use is incompatible with the version of the Broker Service schema on the database. Upgrade the Broker Service to a more recent version.

#### DBOlderVersionThanService

The version of the Broker Service schema on the database is incompatible with the version of the Broker Service currently in use. Upgrade the database schema to a more recent version.

#### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

#### OK

The Broker Service is running and is connected to a database containing a valid schema.

#### PendingFailure

Connectivity between the Broker Service and the database has been lost. This may be a transitory network error, but may indicate a loss of connectivity that requires administrator intervention.

#### Failed

Connectivity between the Broker and the database has been lost for an extended period of time, or has failed due to a configuration problem. The Broker service cannot operate while its connection to the database is unavailable.

## Unknown

The service status cannot be determined.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

### PermissionDenied

You do not have permission to execute this command.

### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

### CommunicationError

There was a problem communicating with the remote service.

### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS>Get-BrokerServiceStatus
```

Get the current status of the Broker Service.

# Get-BrokerSession

Jan 04, 2017

Gets a list of sessions.

## Syntax

```
Get-BrokerSession [-UId] <Int64> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerSession [[-SessionKey] <Guid>] [-AgentVersion <String>] [-ApplicationInUse <String>] [-AppState <SessionAppState>] [-AppStateLastChangeTime <DateTime>] [-AutonomouslyBrokered <Boolean>] [-BrokeringDuration <Int32>] [-BrokeringTime <DateTime>] [-BrokeringUserName <String>] [-BrokeringUserSID <String>] [-CatalogName <String>] [-ClientAddress <String>] [-ClientName <String>] [-ClientPlatform <String>] [-ClientProductId <Int32>] [-ClientVersion <String>] [-ConnectedViaHostName <String>] [-ConnectedViaIP <String>] [-ConnectionMode <ConnectionMode>] [-ControllerDNSName <String>] [-DesktopGroupName <String>] [-DesktopGroupUid <Int32>] [-DesktopKind <DesktopKind>] [-DesktopSID <String>] [-DesktopUid <Int32>] [-DeviceId <String>] [-DNSName <String>] [-EstablishmentDuration <Int32>] [-EstablishmentTime <DateTime>] [-HardwareId <String>] [-Hidden <Boolean>] [-HostedMachineName <String>] [-HostingServerName <String>] [-HypervisorConnectionName <String>] [-ImageOutOfDate <Boolean>] [-InMaintenanceMode <Boolean>] [-IPAddress <String>] [-IsAnonymousUser <Boolean>] [-IsPhysical <Boolean>] [-LaunchedViaHostName <String>] [-LaunchedViaIP <String>] [-LogoffInProgress <Boolean>] [-LogonInProgress <Boolean>] [-MachineName <String>] [-MachineSummaryState <DesktopSummaryState>] [-MachineUid <Int32>] [-Metadata <String>] [-OSType <String>] [-PersistUserChanges <PersistUserChanges>] [-PowerState <PowerState>] [-Protocol <String>] [-ProvisioningType <ProvisioningType>] [-ReceiverIPAddress <String>] [-ReceiverName <String>] [-SecureIcaActive <Boolean>] [-SessionId <Int32>] [-SessionState <SessionState>] [-SessionStateChangeTime <DateTime>] [-SessionSupport <SessionSupport>] [-SessionType <SessionType>] [-StartTime <DateTime>] [-UntrustedUserName <String>] [-UserFullName <String>] [-UserName <String>] [-UserSID <String>] [-UserUPN <String>] [-ApplicationUid <Int32>] [-SharedDesktopUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves sessions matching all the specified criteria. If no parameters are specified this cmdlet enumerates all sessions.

----- BrokerSession Object

The session object returned represents a session on a machine in the site. The session could be for a desktop or application

-- AgentVersion (System.String)

Version of the Citrix Virtual Delivery Agent (VDA) installed on the machine.

-- ApplicationsInUse (System.String[])

List of applications in use in the session. Applications are identified by their administrative name.

-- AppState (Citrix.Broker.Admin.SDK.SessionAppState)

The app state of the session. Valid values are PreLogon, PreLaunched, Active, Desktop, Lingering and NoApps.

-- AppStateLastChangeTime (System.DateTime?)

The time when the session entered the current app state.

-- AutonomouslyBrokered (System.Boolean)

Indicates whether this is an HDX session established by direct connection without being brokered.

-- BrokeringDuration (System.Int32?)

Time taken to broker the session (in milliseconds).

-- BrokeringTime (System.DateTime?)

Time at which the session was brokered.

-- BrokeringUserName (System.String)

The user name of the brokering user.

-- BrokeringUserSID (System.String)

The SID of the brokering user.

-- CatalogName (System.String)

The name of the catalog that the machine hosting the session is assigned to.

-- ClientAddress (System.String)

The IP address of the client connected to the session.

-- ClientName (System.String)

The host name of the client connected to the session.

-- ClientPlatform (System.String)

The name of client platform, as indicated by client product ID.

-- ClientProductId (System.Int32?)

The product ID of the client connected to the session.

-- ClientVersion (System.String)

The version of the Citrix Receiver running on the client connected to the session.

-- ConnectedViaHostName (System.String)

The host name of the incoming connection. This is usually a gateway, router or client.

-- ConnectedViaIP (System.String)

The IP address of the incoming connection This is usually a gateway, router or client.

-- ConnectionMode (Citrix.Broker.Admin.SDK.ConnectionMode?)

The way in which the most recent connection to the session was established. Valid modes are:

- o Brokered: established through the XenDesktop Broker (for example, using StoreFront).
- o Unbrokered: direct connection (for example, using the console or direct RDP/HDX).
- o LeasedConnection: established through the XenDesktop Broker using a connection lease.
- o VdaHighAvailabilityMode: direct connection while VDA in high-availability mode.
- o ThirdPartyBroker: established through a third-party Broker.
- o ThirdPartyBrokerWithLicensing: established and licensed through a third-party Broker.

-- ControllerDNSName (System.String)

The DNS host name of the controller that the session's hosting machine is registered with.

-- DesktopGroupName (System.String)

Name of the desktop group of the machine the session is on.

-- DesktopGroupUid (System.Int32)

UID of the desktop group of the machine the session is on.

-- DesktopKind (Citrix.Broker.Admin.SDK.DesktopKind)

Indicates if the session is shared or private.

-- DesktopSID (System.String)

The Windows SID of the machine the session is on.

-- DesktopUid (System.Int32)

For a desktop session, the unique identifier of the desktop.

-- DeviceId (System.String)

Unique identifier for the client device that has most recently been associated with the session.

-- DNSName (System.String)

The DNS host name of the machine hosting the session.

-- EstablishmentDuration (System.Int32?)

Duration that it took to establish the session.

-- EstablishmentTime (System.DateTime?)

Time at which the session was established.

-- HardwareId (System.String)

Unique identifier for the client hardware that has been most recently associated with the session.

-- Hidden (System.Boolean)

Flag to indicate if the session is currently hidden from the user and not to be reconnected to.

-- HostedMachineName (System.String)

The friendly name of a hosted machine running the session, as used by its hypervisor. This does not necessarily match either the DNS or AD name of the machine.

-- HostingServerName (System.String)

DNS name of the hypervisor that is hosting the machine hosting the session.

-- HypervisorConnectionName (System.String)

The name of the hypervisor connection that the machine hosting the session has been assigned to.

-- ImageOutOfDate (System.Boolean?)

Denotes whether the VM image for a hosted machine is out of date and due to be updated to a new master image when the machine next reboots.

-- InMaintenanceMode (System.Boolean)

Denotes whether the machine hosting the session is in maintenance mode.

-- IPAddress (System.String)

The IP address of the machine hosting the session.

-- IsAnonymousUser (System.Boolean)

Indicates whether the session was established anonymously (without user credentials), in this case a temporary local user account on the machine is used.

-- IsPhysical (System.Boolean)

This value is false if the machine hosting the session can be power managed, and true otherwise

-- LaunchedViaHostName (System.String)

The host name of the StoreFront server used to launch the session.

-- LaunchedViaIP (System.String)

The IP address of the StoreFront server used to launch the session.

-- LogoffInProgress (System.Boolean)

Indicates whether the session is in the process of being logged off.

-- LogonInProgress (System.Boolean)

Indicates whether the session is still executing user logon processing or not.

-- MachineName (System.String)

DNS host name of the machine hosting the session.

-- MachineSummaryState (Citrix.Broker.Admin.SDK.DesktopSummaryState)

The summary state of the machine (will be Unregistered, Disconnected, or InUse)

-- MachineUid (System.Int32)

UID of the machine hosting the session.

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

Map of metadata for this session.

-- OSType (System.String)

A string that can be used to identify the operating system that is running on the machine hosting the session.

-- PersistUserChanges (Citrix.Broker.Admin.SDK.PersistUserChanges)

Describes whether/how the user changes are persisted. Possible values are:

o OnLocal - Persist the user changes locally.

o Discard - Discard user changes.

o OnPvd - Persist user changes on the Citrix Personal vDisk.

-- PowerState (Citrix.Broker.Admin.SDK.PowerState)

The current power state of the machine hosting the session. Possible values are: Unmanaged, Unknown, Unavailable, On, Suspended, TurningOn, TurningOff, Suspending and Resuming.

-- Protocol (System.String)

The protocol that the session is using, can be HDX, RDP, or Console. Console sessions on XenDesktop 5 VDAs appear with a blank protocol.

-- ProvisioningType (Citrix.Broker.Admin.SDK.ProvisioningType)

Describes how the machine hosting the session was provisioned, possible values are:

o Manual: No provisioning.

o PVS: Machine provisioned by PVS (may be physical, blade, VM,...)

o MCS: Machine provisioned by MCS (machine must be VM)

-- ReceiverIPAddress (System.String)

The IP address of the client as supplied by Receiver (for example, StoreFront) when the session was launched, or reconnected.

-- ReceiverName (System.String)

The name of the client as supplied by Receiver (for example, StoreFront) when the session was launched, or reconnected.

-- SecureIcaActive (System.Boolean?)

Indicates whether SecureICA is active on the session.

-- SessionId (System.Int32)

Deprecated. A unique identifier that Remote Desktop Services uses to track the session but it is only unique on that machine.

-- SessionKey (System.Guid)

GUID that provides a unique identifier for this session.

-- SessionState (Citrix.Broker.Admin.SDK.SessionState)

The state of the session. Valid values are Connected, Active or Disconnected.

For a session on a machine with functional level below L7, the additional states PreparingSession, Reconnecting, NonBrokeredSession, Other, and Unknown can also occur.

-- SessionStateChangeTime (System.DateTime)

The time of the most recent state change for the session.

-- SessionSupport (Citrix.Broker.Admin.SDK.SessionSupport)

Indicates if the machine hosting the session supports multiple or single sessions.

-- SessionType (Citrix.Broker.Admin.SDK.SessionType)

Indicates if this is an Application or Desktop session.

-- SmartAccessTags (System.String[])

The Smart Access tags for this session.

-- StartTime (System.DateTime?)

Indicates when the session was started.

-- Uid (System.Int64)

Unique identifier of this session.

-- UntrustedUserName (System.String)

The name of the logged-on user reported directly from the machine (in the form DOMAIN\user). This may be useful where the user is logged in to a non-domain account, however the name cannot be verified and must therefore be considered untrusted.

-- UserFullName (System.String)

The full name of the user.



-- Username (System.String)

The name of the user.

-- UserSID (System.String)

The user's Windows SID.

-- UserUPN (System.String)

The user's User Principal Name

## Related topics

[Disconnect-BrokerSession](#)

[Stop-BrokerSession](#)

[Get-BrokerDesktop](#)

## Parameters

**-Uid**<Int64>

Get session by its Uid.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionKey**<Guid>

Gets session having the specified unique key.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AgentVersion**<String>

Gets sessions with a specific Virtual Desktop Agent version.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationInUse**<String>

Gets sessions running specific applications (identified by their SDK Name property).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AppState**<SessionAppState>

Get sessions by their app state.

Valid values are PreLogon, PreLaunched, Active, Desktop, Lingered and NoApps.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AppStateLastChangeTime**<DateTime>

Get sessions by their app state change time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AutonomouslyBrokered**<Boolean>

Gets sessions according to whether they are autonomously brokered or not. Autonomously brokered sessions are HDX sessions established by direct connection without being brokered.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrokeringDuration**<Int32>

Gets session with a specific time taken to broker. In general, Citrix recommends using -Filter and relative comparisons.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrokeringTime**<DateTime>

Get sessions brokered at a specific time. In general, Citrix recommends using -Filter and relative comparisons.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrokeringUserName**<String>

Get sessions by brokering user.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrokeringUserSID**<String>

Get sessions by brokering user SID.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-CatalogName**<String>

Gets sessions on machines from a specific catalog name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ClientAddress**<String>

Get sessions by client IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Client Name**<String>

Get sessions by client name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Client Platform**<String>

Get sessions by client platform.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

|  |  |
|--|--|
|  |  |
|--|--|

**-ClientProductId<Int32>**

Get sessions by client product ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ClientVersion<String>**

Get sessions by client version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ConnectedViaHostName<String>**

Get sessions by host name of the incoming connection. This is usually a proxy or Citrix Access Gateway server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ConnectedViaIP<String>**

Get sessions by IP address of the incoming connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ConnectionMode**<ConnectionMode>

Gets sessions by the way in which the most recent connection to the session was established.

Valid modes are Brokered, Unbrokered, LeasedConnection, VdaHighAvailabilityMode, ThirdPartyBroker, and ThirdPartyBrokerWithLicensing.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ControllerDNSName**<String>

Gets sessions that are hosted on machines which are registered with a specific controller.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupName**<String>

Gets sessions from a desktop group with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Gets sessions from a desktop group with the specified UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

---

**-DesktopKind**<DesktopKind>

Gets sessions on a desktop of a particular kind.

Valid values are Private and Shared.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopSID**<String>

Get sessions by desktop SID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopUid**<Int32>

Get sessions by desktop Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DeviceId**<String>

Get sessions by client device id.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DNSName<String>**

Gets sessions by their machine's DNS name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-EstablishmentDuration<Int32>**

Gets sessions which took a specific time to establish. In general, Citrix recommends using -Filter and relative comparisons.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-EstablishmentTime<DateTime>**

Gets sessions which became established at a particular time. In general, Citrix recommends using -Filter and relative comparisons.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HardwareId<String>**

Get sessions by client hardware id.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-Hidden**<Boolean>

Get sessions by whether they are hidden or not. Hidden sessions are treated as though they do not exist when brokering sessions; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineName**<String>

Gets sessions by their machine's name as known to its hypervisor.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostingServerName**<String>

Gets sessions hosted by a machine with a specific name of the hosting hypervisor server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionName**<String>

Gets sessions hosted by a machine with a specific name of the hosting hypervisor connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ImageOutOfDate**<Boolean>

Gets sessions hosted by a machine with a specific ImageOutOfDate setting.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InMaintenanceMode**<Boolean>

Gets sessions hosted by a machine with a specific InMaintenanceMode setting.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IPAddress**<String>

Gets sessions hosted by a machine with a specific IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsAnonymousUser**<Boolean>

Gets sessions depending on whether they were established anonymously (\$true) or not (\$false). An anonymous session is established without user credentials and a temporary local user account is used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsPhysical**<Boolean>

Gets sessions hosted on machines where the flag indicating if the machine can be power managed by the Citrix Broker Service matches the requested value. Where the power state of the machine cannot be controlled, specify \$true, otherwise \$false.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LaunchedViaHostName**<String>

Get sessions by the host name of the Web Interface server from which a user launches a session.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LaunchedViaIP**<String>

Get sessions by the IP address of the Web Interface server from which a user launches a session.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LogoffInProgress**<Boolean>

Gets sessions by whether they are in the process of being logged off or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LogonInProgress**<Boolean>

Gets sessions by whether they are still executing user logon processing or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName**<String>

Gets sessions by their machine name (in the form DOMAIN\machine).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineSummaryState**<DesktopSummaryState>

Gets sessions on a machine with a specific summary state.

Valid values are Off, Unregistered, Available, Disconnected, Preparing, and InUse.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineUid**<Int32>

Gets sessions on a machine with the specified UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSType**<String>

Gets sessions with a specific type of operating system.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PersistUserChanges**<PersistUserChanges>

Gets sessions where the user changes are persisted in a particular manner. Values can be:

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the Pvd.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PowerState**<PowerState>

Gets sessions on machines in the specified power state.

Valid values are Unmanaged, Unknown, Unavailable, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Protocol**<String>

Get sessions by connection protocol. Valid values are HDX, RDP, or Console.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ProvisioningType**<ProvisioningType>

Gets sessions hosted on machines provisioned in a particular manner. Values can be:

- o Manual - No automated provisioning.
- o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).
- o MCS - Machine provisioned by MCS (machine must be VM).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReceiverIPAddress**<String>

Gets sessions with the specified client IP address supplied by Receiver (for example, StoreFront) when the session was launched, or reconnected.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReceiverName**<String>

Gets sessions with the specified client name supplied by Receiver (for example, StoreFront) when the session was launched, or reconnected.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaActive**<Boolean>

Get sessions by their use of SecureICA.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionId**<Int32>

Deprecated.

Gets sessions by session ID, a unique identifier that Remote Desktop Services uses to track the session but it is only unique on that machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionState**<SessionState>

Get sessions by their state.

Valid values are Other, PreparingNewSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession, and Unknown.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? |       |

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionStateChangeTime**<DateTime>

Get sessions by their last state change time. In general, Citrix recommends using -Filter and relative comparisons.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionSupport**<SessionSupport>

Gets sessions hosted on machines which support the required pattern of sessions. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionType**<SessionType>

Get sessions by their type.

Valid values are Application and Desktop.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartTime**<DateTime>

Get sessions by their start time. In general, Citrix recommends using -Filter and relative comparisons.



|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UntrustedUserName**<String>

Gets sessions by the untrusted user name reported directly from the machine (in the form DOMAIN\user).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserFullName**<String>

Gets sessions by user's full name (usually 'first-name last-name').

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserName**<String>

Get sessions by user name (in the form DOMAIN\user).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserSID**<String>

Get sessions by user's Windows SID.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserUPN**<String>

Gets sessions by user's User Principal Name (in the form user@domain).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationUid**<Int32>

Get sessions running the application with the specified Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SharedDesktopUid**<Int32>

Get sessions by SharedDesktop Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.Session

Returns sessions matching the specified criteria.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerSession  
Enumerate all sessions.

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerSession -UserName MyDomain\MyAccount -SessionState Disconnected
Get all disconnected sessions for a specific user.
```

# Get-BrokerSessionLinger

Jan 04, 2017

Gets one or more session lingering settings.

## Syntax

```
Get-BrokerSessionLinger [-DesktopGroupId] <Int32> [-Property <String[]>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Get-BrokerSessionLinger [[-DesktopGroupName] <String>] [-AssociatedUserFullName <String>] [-AssociatedUserName
<String>] [-AssociatedUserUPN <String>] [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-
MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate
<TimeSpan>] [-UserFilterEnabled <Boolean>] [-UserSID <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-
Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Get-BrokerSessionLinger cmdlet is used to enumerate desktop group session linger settings that match all of the supplied criteria.

Without parameters, Get-BrokerSessionLinger gets all the session linger settings that have been created. You can also use the parameters of Get-BrokerSessionLinger to filter the results to just the desktop group you're interested in.

Note that each desktop group can only have a single session linger setting. Session lingering only applies to application sessions.

See about\_Broker\_Filtering for information about advanced filtering options.

----- BrokerSessionLinger Object

The session linger object returned represents a session linger setting in a desktop group.

-- AssociatedUserFullNames (System.String[])

List of associated users (full names). Associated users is the list of users who are given access using the pre-launch/user mapping filter.

-- AssociatedUserNames (System.String[])

List of associated users (SAM names). Associated users is the list of users who are given access using the pre-launch/user mapping filter.

-- AssociatedUserUPNs (System.String[])

List of associated users (user principle names). Associated users is the list of users who are given access using the pre-launch/user mapping filter.

-- DesktopGroupName (System.String)

Name of the associated desktop group.

-- DesktopGroupUid (System.Int32)

Uid of the associated desktop group.

-- Enabled (System.Boolean)

Specifies whether or not session lingering is enabled for the desktop group.

-- MaxAverageLoadThreshold (System.Int32)

Specifies the average load threshold across the desktop group. After this threshold is hit lingering sessions will be terminated to reduce average load across the group. Sessions that have been lingering the longest will be chosen first.

-- MaxLoadPerMachineThreshold (System.Int32)

Specifies the maximum load threshold per machine in the desktop group. After this threshold is hit lingering sessions on loaded machines will be terminated to reduce load. Sessions that have been lingering the longest will be chosen first.

-- MaxTimeBeforeDisconnect (System.TimeSpan)

Specifies the maximum time by when a lingering session will be disconnected. The disconnect timer cannot be greater than the terminate timer. When the disconnect timer is same as the terminate timer, the session will be directly be terminated. The default value is 15 minutes. A value of 0 disables the disconnect timer.

-- MaxTimeBeforeTerminate (System.TimeSpan)

Specifies the maximum time by when a lingering session will be terminated. When the disconnect timer is same as the terminate timer, the session will be directly be terminated. The default value is 8 hours. A value of 0 disables the terminate timer.

-- UserFilterEnabled (System.Boolean)

Indicates if linger-specific user filter is enabled.

## Related topics

[New-BrokerSessionLinger](#)

[Set-BrokerSessionLinger](#)

[Remove-BrokerSessionLinger](#)

## Parameters

**-DesktopGroupUid**<Int32>

Gets session linger setting that is associated with the specified desktop group Uid.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupName<String>**

Gets session linger setting that is associated with the specified desktop group name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserFullName<String>**

Gets session linger settings with an associated user identified by their full name (usually 'first-name last-name'). If the 'UserFilterEnabled' property is true then access to the session linger is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserName<String>**

Gets session linger settings with an associated user identified by their user name (in the form 'domain\user'). If the 'UserFilterEnabled' property is true then access to the session linger is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserUPN<String>**

Gets session linger settings with an associated user identified by their user principle name (in the form 'user@domain'). If the 'UserFilterEnabled' property is true then access to the session linger is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Gets only the session linger settings that have the specified value for whether the setting is enabled.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxAverageLoadThreshold**<Int32>

Gets only the session linger settings that have the specified average load threshold.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxLoadPerMachineThreshold**<Int32>

Gets only the session linger settings that have the specified maximum load threshold per machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxTimeBeforeDisconnect**<TimeSpan>

Gets only the session linger settings that have the specified idle disconnect time.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-MaxTimeBeforeTerminate**<TimeSpan>

Gets only the session linger settings that have the specified idle terminate time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserFilterEnabled**<Boolean>

Gets only session linger settings whose user filter is in the specified state.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserSID**<String>

Gets only session linger settings with their accessibility restricted to include the specified user.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | False |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.SessionLinger

Get-BrokerSessionLinger returns an object for each session linger setting it gets.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerSessionLinger -DesktopGroupName "test"

Returns the session linger settings associated with the desktop group named 'test'.

# Get-BrokerSessionPreLaunch

Jan 04, 2017

Gets one or more session pre-launch settings.

## Syntax

```
Get-BrokerSessionPreLaunch [-DesktopGroupUId] <Int32> [-Property <String[]>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Get-BrokerSessionPreLaunch [[-DesktopGroupName] <String>] [-AssociatedUserFullName <String>] [-AssociatedUserName
<String>] [-AssociatedUserUPN <String>] [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-
MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate
<TimeSpan>] [-UserFilterEnabled <Boolean>] [-UserSID <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-
Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Get-BrokerSessionPreLaunch cmdlet is used to enumerate desktop group session pre-launch settings that match all of the supplied criteria.

Without parameters, Get-BrokerSessionPreLaunch gets all the session pre-launch settings that have been created. You can also use the parameters of Get-BrokerSessionPreLaunch to filter the results to just the desktop group you're interested in.

Note that each desktop group can only have a single session pre-launch setting. Session pre-launch only applies to application sessions.

See about\_Broker\_Filtering for information about advanced filtering options.

----- BrokerSessionPreLaunch Object

The session pre-launch object returned represents a session pre-launch setting in a desktop group.

-- AssociatedUserFullNames (System.String[])

List of associated users (full names). Associated users is the list of users who are given access using the pre-launch/user mapping filter.

-- AssociatedUserNames (System.String[])

List of associated users (SAM names). Associated users is the list of users who are given access using the pre-launch/user mapping filter.

-- AssociatedUserUPNs (System.String[])

List of associated users (user principle names). Associated users is the list of users who are given access using the pre-launch/user mapping filter.

-- DesktopGroupName (System.String)

Name of the associated desktop group.

-- DesktopGroupUid (System.Int32)

Uid of the associated desktop group.

-- Enabled (System.Boolean)

Specifies whether or not session pre-launch is enabled for the desktop group.

-- MaxAverageLoadThreshold (System.Int32)

Specifies the average load threshold across the desktop group. After this threshold is hit pre-launched sessions will be terminated to reduce average load across the group. Sessions that have been pre-launched the longest will be chosen first.

-- MaxLoadPerMachineThreshold (System.Int32)

Specifies the maximum load threshold per machine in the desktop group. After this threshold is hit pre-launched sessions on loaded machines will be terminated to reduce load. Sessions that have been pre-launched the longest will be chosen first.

-- MaxTimeBeforeDisconnect (System.TimeSpan)

Specifies the maximum time by when a pre-launched session will be disconnected. The disconnect timer cannot be greater than the terminate timer. When the disconnect timer is same as the terminate timer, the session will be directly be terminated. The default value is 15 minutes. A value of 0 disables the disconnect timer.

-- MaxTimeBeforeTerminate (System.TimeSpan)

Specifies the maximum time by when a pre-launched session will be terminated. When the disconnect timer is same as the terminate timer, the session will be directly be terminated. The default value is 8 hours. A value of 0 disables the terminate timer.

-- UserFilterEnabled (System.Boolean)

Indicates if pre-launch-specific user filter is enabled.

## Related topics

[New-BrokerSessionPreLaunch](#)

[Set-BrokerSessionPreLaunch](#)

[Remove-BrokerSessionPreLaunch](#)

## Parameters

**-DesktopGroupUid**<Int32>

Gets session pre-launch setting that is associated with the specified desktop group Uid.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupName<String>**

Gets session pre-launch setting that is associated with the specified desktop group name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserFullName<String>**

Gets session pre-launch settings with an associated user identified by their full name (usually 'first-name last-name'). If the 'UserFilterEnabled' property is true then access to the session pre-launch is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserName<String>**

Gets session pre-launch settings with an associated user identified by their user name (in the form 'domain\user'). If the 'UserFilterEnabled' property is true then access to the session pre-launch is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserUPN<String>**

Gets session pre-launch settings with an associated user identified by their user principle name (in the form 'user@domain'). If the 'UserFilterEnabled' property is true then access to the session pre-launch is restricted to those users only, otherwise access is unrestricted (but always subject to other policy rules).

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Gets only the session pre-launch settings that have the specified value for whether the setting is enabled.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxAverageLoadThreshold**<Int32>

Gets only the session pre-launch settings that have the specified average load threshold.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxLoadPerMachineThreshold**<Int32>

Gets only the session pre-launch settings that have the specified maximum load threshold per machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxTimeBeforeDisconnect**<TimeSpan>

Gets only the session pre-launch settings that have the specified idle disconnect time.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |



|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-MaxTimeBeforeTerminate**<TimeSpan>

Gets only the session pre-launch settings that have the specified idle terminate time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserFilterEnabled**<Boolean>

Gets only session pre-launch settings whose user filter is in the specified state.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserSID**<String>

Gets only session pre-launch settings with their accessibility restricted to include the specified user.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | False |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.SessionPreLaunch

Get-BrokerSessionPreLaunch returns an object for each session pre-launch setting it gets.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerSessionPreLaunch -DesktopGroupName "test"

Returns the session pre-launch settings associated with the desktop group named 'test'.

# Get-BrokerSharedDesktop

Jan 04, 2017

Get shared desktops configured for this site.

## Syntax

```
Get-BrokerSharedDesktop [-UId] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerSharedDesktop [[-MachineName] <String>] [-AgentVersion <String>] [-ControllerDNSName <String>] [-DesktopGroupUId <Int32>] [-DNSName <String>] [-HostedMachineId <String>] [-HostedMachineName <String>] [-HostingServerName <String>] [-HypervisorConnectionUId <Int32>] [-InMaintenanceMode <Boolean>] [-IPAddress <String>] [-LastDeregistrationReason <DeregistrationReason>] [-LastDeregistrationTime <DateTime>] [-LastHostingUpdateTime <DateTime>] [-OSType <String>] [-OSVersion <String>] [-PowerState <PowerState>] [-RegistrationState <RegistrationState>] [-SID <String>] [-Tag <String>] [-WillShutdownAfterUse <Boolean>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet is deprecated, please use the Get-BrokerMachine cmdlet instead.

Retrieve shared desktops matching the specified criteria. If no parameters are specified, all shared desktops are enumerated.

Get-BrokerSharedDesktop returns configuration information only for shared desktops (a DesktopKind of 'Shared').

For information about advanced filtering options, see about\_Broker\_Filtering; for information about desktops, see about\_Broker\_Desktops.

### ----- BrokerSharedDesktop Object

Shared desktops are desktops that are assigned randomly to users upon connection from a pool of available machines.

-- AgentVersion (System.String)

Version of the Citrix Virtual Delivery Agent (VDA) installed on the desktop.

-- ControllerDNSName (System.String)

The DNS host name of the controller that the desktop is registered to.

-- DesktopGroupUId (System.Int32)

UId of the desktop group the desktop has been assigned to.

-- DNSName (System.String)

The DNS host name of the desktop.

-- HostedMachineId (System.String)

Unique ID within the hosting unit of the target managed desktop.

-- HostedMachineName (System.String)

The friendly name of a hosted desktop as used by its hypervisor. This is not necessarily the DNS name of the desktop.

-- HostingServerName (System.String)

DNS name of the hypervisor that is hosting the desktop if managed.

-- HypervisorConnectionUid (System.Int32?)

The UID of the hypervisor connection that the desktop has been assigned to, if managed.

-- InMaintenanceMode (System.Boolean)

Denotes whether the desktop is in maintenance mode.

-- IPAddress (System.String)

The IP address of the desktop.

-- LastDeregistrationReason (Citrix.Broker.Admin.SDK.DeregistrationReason?)

The reason for the last deregistration of the desktop with the broker. Possible values are:

AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

-- LastDeregistrationTime (System.DateTime?)

Time of the last deregistration of the desktop from the controller.

-- LastHostingUpdateTime (System.DateTime?)

Time of last update to any hosting data for this desktop reported by the hypervisor connection.

-- MachineName (System.String)

DNS host name of the machine associated with the desktop.

-- OSType (System.String)

A string that can be used to identify the operating system that is running on the desktop.

-- OSVersion (System.String)

A string that can be used to identify the version of the operating system running on the desktop, if known.

-- PowerState (Citrix.Broker.Admin.SDK.PowerState)

The current power state of the desktop. Possible values are: Unmanaged, Unknown, Unavailable, Off, On, Suspended,

TurningOn, TurningOff, Suspending, resuming.

-- RegistrationState (Citrix.Broker.Admin.SDK.RegistrationState)

Indicates the registration state of the desktop. Possible values are: Unregistered, Initializing, Registered, AgentError.

-- SID (System.String)

The security identifier of the shared desktop.

-- Uid (System.Int32)

The uid of the shared desktop.

-- WillShutdownAfterUse (System.Boolean)

Flag indicating whether this desktop is tainted and will be shutdown after all sessions on the desktop have ended. This flag should only ever be true on power managed, single-session desktops.

Note: The desktop will not shut down if it is in maintenance mode, but will shut down after the desktop is taken out of maintenance mode.

## Related topics

[Set-BrokerSharedDesktop](#)

## Parameters

**-Uid**<Int32>

Gets desktops by Uid.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName**<String>

Gets desktops by machine name (in the form 'domain\machine').

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AgentVersion**<String>

Gets desktops with a specific Virtual Desktop Agent version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ControllerDNSName<String>**

Gets desktops by the DNS name of the controller they are registered with.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid<Int32>**

Gets desktops from a desktop group with a specific Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DNSName<String>**

Gets desktops by DNS name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineId<String>**

Gets desktops by the machine id known to the hypervisor.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineName**<String>

Gets desktops by the machine name known to the hypervisor.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostingServerName**<String>

Gets desktops by the name of the hosting hypervisor server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionUid**<Int32>

Gets desktops by the uid of the hosting hypervisor connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InMaintenanceMode**<Boolean>

Gets desktops by the InMaintenanceMode setting.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IPAddress**<String>

Gets desktops by IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastDeregistrationReason**<DeregistrationReason>

Gets desktops whose broker last recorded a specific deregistration reason.

Valid values are \$null, AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastDeregistrationTime**<DateTime>

Gets desktops by the time that they were last deregistered.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastHostingUpdateTime**<DateTime>

Gets desktops by the time that the hosting information was last updated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSType**<String>

Gets desktops by the type of operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSVersion**<String>

Gets desktops by the version of the operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PowerState**<PowerState>

Gets desktops by power state.

Valid values are Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RegistrationState**<RegistrationState>

Gets desktops by registration state.

Valid values are Registered, Unregistered, and AgentError.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SID**<String>

Gets desktops by machine SID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Tag**<String>

Get desktops tagged with the given tag.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WillShutDownAfterUse**<Boolean>

Gets desktops depending on whether they shutdown after use or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|               |                                                         |
|---------------|---------------------------------------------------------|
| Required?     | false                                                   |
| Default Value | The default sort order is by name or unique identifier. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.SharedDesktop

Get-BrokerSharedDesktop returns an object for each matching shared desktop.

## Notes

To compare dates/times, use `-Filter` and relative comparisons. For more information, see `about_Broker_Filtering`.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerSharedDesktop -HostingServerName BigServer12*
```

Get all shared desktops hosted by the hypervisor server BigServer12.

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerSharedDesktop -OSType 'Windows XP*' | ft -a MachineName,OSType,OSVersion
```

List all shared desktops running Windows XP, including the machine name and OS details.

# Get-BrokerSite

Jan 04, 2017

Gets the current XenDesktop broker site.

## Syntax

```
Get-BrokerSite [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Get-BrokerSite cmdlet gets the current broker site.

The broker site is a top-level, logical representation of the XenDesktop site, from the perspective of the brokering services running within the site. It defines various site-wide default attributes used by the brokering services.

A XenDesktop installation has only a single broker site instance.

### ----- BrokerSite Object

The BrokerSite object represents logical representation of the XenDesktop site. It contains the following properties:

-- BaseOU (System.Guid?)

The objectGUID property identifying the base OU in Active Directory used for desktop registrations.

-- BrokerServiceGroupUid (System.Guid)

The Uid for the Broker Service Group.

-- ColorDepth (Citrix.Broker.Admin.SDK.ColorDepth)

The default color depth for new desktop groups.

-- ConfigurationServiceGroupUid (System.Guid?)

The Uid for the Configuration Service Group.

-- ConnectionLeasingEnabled (System.Boolean?)

The indicator for connection leasing active.

-- DesktopGroupIconUid (System.Int32)

The default desktop icon used for new desktop groups.

-- DnsResolutionEnabled (System.Boolean)

The setting to configure whether numeric IP address or the DNS name to be present in the ICA file.

-- LicensedSessionsActive (System.Int32?)

The count of active licensed session.

-- LicenseEdition (System.String)

The license edition for session brokering.

-- LicenseGraceSessionsRemaining (System.Int32?)

The count of License Grace Session Remaining

-- LicenseModel (Citrix.Broker.Admin.SDK.LicenseModel?)

The licensing model in use. Values can be 'Concurrent' or 'UserDevice'

-- LicenseServerName (System.String)

The DNS for License Server Name

-- LicenseServerPort (System.Int32)

The port for the License Server

-- LicensingBurnIn (System.String)

The date for the license to end in yyyy.MMdd format

-- LicensingBurnInDate (System.DateTime?)

The date for the license to end

-- LicensingGraceHoursLeft (System.Int32?)

The number of grace hours left after license expiry

-- LicensingGracePeriodActive (System.Boolean?)

The indicator for licensing grace period active

-- LicensingOutOfBoxGracePeriodActive (System.Boolean?)

The indicator for licensing out of the box grace period active

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

The metadata for this command.

-- Name (System.String)

The name of the site

-- SecureIcaRequired (System.Boolean)

The default SecureICA usage requirements for new desktop groups.

-- TrustRequestsSentToTheXmlServicePort (System.Boolean)

The XML Service trust settings.



## Related topics

[Set-BrokerSite](#)

[Get-BrokerIcon](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

Citrix.Broker.Admin.SDK.Site

Get-BrokerSite returns the single broker site instance.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerSite  
Gets the current broker site.
```

# Get-BrokerTag

Jan 04, 2017

Gets one or more tags.

## Syntax

```
Get-BrokerTag [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerTag [[-Name] <String>] [-Metadata <String>] [-UUID <Guid>] [-DesktopUid <Int32>] [-DesktopGroupUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets tags that match all of the supplied criteria.

----- BrokerTag Object

The BrokerTag object represents a single instance of a Tag associated to other objects. It contains the following properties:

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

The metadata for this command.

-- Name (System.String)

The name of the Tag

-- Uid (System.Int32)

The Uid of the Tag

-- UUID (System.Guid)

The UUID associated to the Tag

## Related topics

[Add-BrokerTag](#)

[New-BrokerTag](#)

[Remove-BrokerTag](#)

[Rename-BrokerTag](#)

## Parameters

**-Uid**<Int32>

Gets the tag identified by Uid

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Name**<String>

Gets tags that match the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UUID**<Guid>

Gets tags associated with a given UUID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopUid**<Int32>

Gets tags associated with a Desktop.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid<Int32>**

Gets tags associated with a desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount<SwitchParameter>**

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount<Int32>**

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip<Int32>**

Skips the specified number of records before returning results. Also reduces the count returned by -

ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None Input cannot be piped to this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.Tag

Returns matching tags.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Get-BrokerTag  
Enumerate all tags

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerTag -Uid 5  
Get a single specific tag with a Uid of 5.

----- **EXAMPLE 3** -----

C:\PS> Get-BrokerTag -DesktopUid 1  
Get tags associated with Desktop 1.

# Get-BrokerUnconfiguredMachine

Jan 04, 2017

Gets machines that have registered but are not yet configured in this site.

## Syntax

```
Get-BrokerUnconfiguredMachine -SID <String> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerUnconfiguredMachine [[-MachineName] <String>] [-AgentVersion <String>] [-ControllerDNSName <String>] [-DNSName <String>] [-FunctionalLevel <FunctionalLevel>] [-LastDeregistrationTime <DateTime>] [-OSType <String>] [-OSVersion <String>] [-SessionSupport <SessionSupport>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieve machines matching the specified criteria, where the machine has registered with a controller in the site but the machine has not yet been configured to be part of the site. If no parameters are specified, this cmdlet enumerates all such machines.

See about\_Broker\_Filtering for information about advanced filtering options, and about\_Broker\_Machines for background information about machines.

----- BrokerUnconfiguredMachine Object

An unconfigured machine is a machine that has registered with the site but is not configured in either a desktop group or a catalog.

-- AgentVersion (System.String)

Version of the Citrix Virtual Delivery Agent (VDA) installed on the unconfigured machine.

-- ControllerDNSName (System.String)

The DNS name of the controller that the unconfigured machine is registered with.

-- DNSName (System.String)

The DNS name of the unconfigured machine.

-- FunctionalLevel (Citrix.Broker.Admin.SDK.FunctionalLevel?)

The functional level of the unconfigured machine. This is determined by the version of the Citrix VDA software installed on the machine.

-- LastDeregistrationTime (System.DateTime?)

The time when the unconfigured machine last deregistered with the Citrix Broker Service.

-- MachineName (System.String)

The machine name of the unconfigured machine in the form domain\machine.

-- OSType (System.String)

The type of operating system installed on the unconfigured machine.

-- OSVersion (System.String)

The version of the operating system installed on the unconfigured machine.

-- SessionSupport (Citrix.Broker.Admin.SDK.SessionSupport?)

The session support of the unconfigured machine. Valid values are:

SingleSession, MultiSession

-- SID (System.String)

Security identifier of the unconfigured machine.

Related topics

[Get-BrokerMachine](#)

Parameters

**-SID**<String>

Gets machines by their machine SID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName**<String>

Gets machines by their machine name (in the form domain\machine).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AgentVersion**<String>

Gets machines with a specific Virtual Desktop Agent version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ControllerDNSName**<String>

Gets machines by the DNS name of the controller they are registered with.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DNSName**<String>



Gets machines by their DNS name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-FunctionalLevel**<FunctionalLevel>

Gets machines with a specific FunctionalLevel.

Valid values are L5, L7, L7\_6

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastDeregistrationTime**<DateTime>

Gets machines by the time that they were last deregistered.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSType**<String>

Gets machines by the type of operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSVersion**<String>

Gets machines by the version of the operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionSupport**<SessionSupport>

Gets machines that have the specified session capability. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

#### **-Filter**<String>

Gets records that match a PowerShell style filter expression. See `about_Broker_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

#### Input Type

None You cannot pipe input into this cmdlet.

#### Return Values

Citrix.Broker.Admin.SDK.UnconfiguredMachine

`Get-BrokerUnconfiguredMachine` returns an object for each matching machine

#### Notes

It is generally better to compare dates and times using `-Filter` and relative comparisons. See `about_Broker_Filtering` and the examples in this topic for more information.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerUnconfiguredMachine -Filter { ControllerDNSName -eq 'Controller1.domain.com' -and OSType -eq 'Windows XP Service Pack 3' }
```

This command returns all unconfigured XP SP3 machines which are registered with the controller called 'Controller1.domain.com'.

# Get-BrokerUser

Jan 04, 2017

Gets broker users configured for this site.

## Syntax

```
Get-BrokerUser -SID <String> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerUser [[-Name] <String>] [-FullName <String>] [-UPN <String>] [-ApplicationUid <Int32>] [-SessionLingerDesktopGroupUid <Int32>] [-SessionPreLaunchDesktopGroupUid <Int32>] [-MachineUid <Int32>] [-PrivateDesktopUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieve broker users matching the specified criteria. If no parameters are specified this cmdlet enumerates all broker users.

For information about advanced filtering options, see [about\\_Broker\\_Filtering](#).

----- BrokerUser Object

The BrokerUser object represents a single instance of an user. It contains the following properties:

-- FullName (System.String)

The full name of an user

-- Name (System.String)

The name of an user

-- SID (System.String)

The SID of an user

-- UPN (System.String)

The UPN of an user

## Related topics

[Add-BrokerUser](#)

[Remove-BrokerUser](#)

## Parameters

**-SID**<String>

Gets the broker user with the specified SID property value.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Name<String>**

Gets the broker user with the specified Name property.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-FullName<String>**

Gets the broker user with the specified FullName property.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UPN<String>**

Gets the broker user with the specified UPN property value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationUid<Int32>**

Gets broker users associated with the application with the specified Uid.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-SessionLingerDesktopGroupUid<Int32>**

Gets broker users associated with the desktop group session linger settings with the specified Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionPreLaunchDesktopGroupUid<Int32>**

Gets broker users associated with the desktop group session pre-launch settings with the specified Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineUid<Int32>**

Gets broker users associated with the broker machine with the specified Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PrivateDesktopUid<Int32>**

Gets broker users associated with the private desktop with the specified Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.User

Get-BrokerUser returns an object for each matching broker user.

## Examples

### ----- EXAMPLE 1 -----

```
Get-BrokerUser DOMAIN7\*
```

Get all broker user objects with names matching the supplied pattern

### ----- EXAMPLE 2 -----

```
$pdt = Get-BrokerPrivateDesktop DOMAIN\MACHINENAME\*
```

```
Get-BrokerUser -PrivateDesktopUid $pdt.Uid
```

Get all broker user objects added to the specified private desktop

# Group-BrokerDesktop

Jan 04, 2017

Groups and counts desktops with the same value for a specified property.

## Syntax

```
Group-BrokerDesktop [-UId <Int32> -Property <String> [-AdminAddress <String>] [<CommonParameters>]
```

```
Group-BrokerDesktop -Property <String> [[-MachineName] <String>] [-AgentVersion <String>] [-ApplicationInUse <String>] [-AssignedClientName <String>] [-AssignedIPAddress <String>] [-AssociatedUserFullName <String>] [-AssociatedUserName <String>] [-AssociatedUserUPN <String>] [-AutonomouslyBrokered <Boolean>] [-CatalogName <String>] [-CatalogUid <Int32>] [-ClientAddress <String>] [-ClientName <String>] [-ClientVersion <String>] [-ColorDepth <ColorDepth>] [-ConnectedViaHostName <String>] [-ConnectedViaIP <String>] [-ControllerDNSName <String>] [-DeliveryType <DeliveryType>] [-Description <String>] [-DesktopCondition <String>] [-DesktopGroupName <String>] [-DesktopGroupUid <Int32>] [-DesktopKind <DesktopKind>] [-DeviceId <String>] [-DNSName <String>] [-FunctionalLevel <FunctionalLevel>] [-HardwareId <String>] [-HostedMachineId <String>] [-HostedMachineName <String>] [-HostingServerName <String>] [-HypervisorConnectionName <String>] [-HypervisorConnectionUid <Int32>] [-IconUid <Int32>] [-ImageOutOfDate <Boolean>] [-InMaintenanceMode <Boolean>] [-IPAddress <String>] [-IsAssigned <Boolean>] [-IsPhysical <Boolean>] [-LastConnectionFailure <ConnectionFailureReason>] [-LastConnectionTime <DateTime>] [-LastConnectionUser <String>] [-LastDeregistrationReason <DeregistrationReason>] [-LastDeregistrationTime <DateTime>] [-LastErrorReason <String>] [-LastErrorTime <DateTime>] [-LastHostingUpdateTime <DateTime>] [-LaunchedViaHostName <String>] [-LaunchedViaIP <String>] [-MachineInternalState <MachineInternalState>] [-MachineUid <Int32>] [-OSType <String>] [-OSVersion <String>] [-PersistUserChanges <PersistUserChanges>] [-PowerActionPending <Boolean>] [-PowerState <PowerState>] [-Protocol <String>] [-ProvisioningType <ProvisioningType>] [-PublishedApplication <String>] [-PublishedName <String>] [-PvdStage <PvdStage>] [-RegistrationState <RegistrationState>] [-SecureIcaActive <Boolean>] [-SecureIcaRequired <Boolean>] [-SessionHidden <Boolean>] [-SessionId <Int32>] [-SessionState <SessionState>] [-SessionStateChangeTime <DateTime>] [-SessionUid <Int64>] [-SessionUserName <String>] [-SessionUserSID <String>] [-SID <String>] [-SmartAccessTag <String>] [-StartTime <DateTime>] [-SummaryState <DesktopSummaryState>] [-Tag <String>] [-WillShutdownAfterUse <Boolean>] [-ApplicationUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet is now deprecated, please use Group-BrokerMachine.

Filters desktops using the specified criteria, then groups and counts matching desktops with the same value for a particular property. The number of desktops in the group, and the property value for the group, is output. For example:

```
C:\PS> Group-BrokerDesktop -Property SummaryState
```

```
Count Name
```

```
-----
```

```
43 Available
```

```
17 InUse
```

```
3 Disconnected
```

Filtering supports the same options as the Get-BrokerDesktop cmdlet, and allows filtering on both desktop and session properties.

Group-BrokerDesktop is similar to the standard PowerShell Group-Object, but is faster than piping the output of Get-BrokerDesktop into Group-Object when working with many desktops.

Note that all session information properties for multi-session desktops is always \$null, this means that it is not possible to group these desktops by session information using this command. Use Get-BrokerSession to get information on all current sessions.

Also note that the MaxRecordCount, ReturnTotalRecordCount, Skip, and SortBy parameters apply to GroupInfo records output rather than the filtered desktops.

## Related topics

[Get-BrokerDesktop](#)

[Group-Object](#)

[Parameters](#)

**-UId**<Int32>

Gets desktops with a specific UID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property<String>**

Selects the property by which matching desktops are grouped.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName<String>**

Gets desktops with a specific machine name (in the form 'domain\machine').

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AgentVersion<String>**

Gets desktops with a specific Citrix Virtual Delivery Agent version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationInUse<String>**

Gets desktops running a specified published application (identified by browser name).

String comparisons are case-insensitive.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedClientName<String>**

Gets desktops assigned to a specific client name.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedIPAddress**<String>

Gets desktops assigned to a specific IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserFullName**<String>

Gets desktops with an associated user identified by their full name (usually in the form 'first-name last-name').

Associated users are the current user for shared desktops, and the assigned users for private desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserName**<String>

Gets desktops with an associated user identified by their user name (in the form 'domain\user').

Associated users are the current user for shared desktops, and the assigned users for private desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserUPN**<String>

Gets desktops with an associated user identified by their User Principle Name (in the form 'user@domain').

Associated users are the current user for shared desktops, and the assigned users for private desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AutonomouslyBrokered**<Boolean>

Gets desktops according to whether their current session is autonomously brokered or not. Autonomously brokered sessions are HDX sessions established by

direct connection without being brokered.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CatalogName**<String>

Gets desktops from the catalog with the specific name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CatalogUid**<Int32>

Gets desktops from a catalog with a specific UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ClientAddress**<String>

Gets desktops with a specific client IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ClientName**<String>

Gets desktops with a specific client name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ClientVersion**<String>

Gets desktops with a specific client version.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ColorDepth**<ColorDepth>

Gets desktops configured with a specific color depth.

Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ConnectedViaHostName**<String>

Gets desktops with a specific host name of the incoming connection. This is usually a proxy or Citrix Access Gateway server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ConnectedViaIP**<String>

Gets desktops with a specific IP address of the incoming connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ControllerDNSName**<String>

Gets desktops with a specific DNS name of the controller they are registered with.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DeliveryType**<DeliveryType>

Gets desktops of a particular delivery type.

Valid values are AppsOnly, DesktopsOnly, DesktopsAndApps

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Get desktops with a specific description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopCondition**<String>

Gets desktop with an outstanding desktop condition condition.

Valid values are:

- o CPU: Indicates the machine has high CPU usage
- o ICALatency: Indicates the network latency is high
- o UPMLogonTime: Indicates that the profile load time was high

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupName**<String>

Gets desktops from a desktop group with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Gets desktops from a desktop group with the specified UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopKind**<DesktopKind>



Deprecated: Use AllocationType parameter.

Gets desktops of a particular kind.

Valid values are Private, Shared.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DeviceId**<String>

Gets desktops with a specific client device ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DNSName**<String>

Get desktops with a specific DNS name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-FunctionalLevel**<FunctionalLevel>

Gets desktops with a specific FunctionalLevel.

Valid values are L5, L7, L7\_6

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HardwareId**<String>

Gets desktops with a specific client hardware ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineId**<String>

Gets desktops with a specific machine ID known to the hypervisor.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineName**<String>

Gets desktops with a specific machine name known to the hypervisor.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostingServerName**<String>

Gets desktops with a specific name of the hosting hypervisor server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionName**<String>

Gets desktops with a specific name of the hosting hypervisor connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionUid**<Int32>

Gets desktops with a specific UID of the hosting hypervisor connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconUid**<Int32>

Gets desktops with a specific configured icon. Note that desktops with a null IconUid use the icon of the desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ImageOutOfDate**<Boolean>

Gets desktops if they have an ImageOutOfDate flag.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InMaintenanceMode**<Boolean>

Gets desktops with a specific InMaintenanceMode setting.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IPAddress**<String>

Gets desktops with a specific IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsAssigned**<Boolean>

Gets desktops according to whether they are assigned or not. Desktops may be assigned to one or more users or groups, a client IP address or a client endpoint name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsPhysical**<Boolean>

Specifies if machines in the catalog can be power managed by the Citrix Broker Service. Where the power state of the machine cannot be controlled, specify \$true, otherwise \$false. Can only be specified together with a provisioning type of Pvs or Manual, or if used with the deprecated CatalogKind parameter only with Pvs or PvsPvd catalog kinds.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastConnectionFailure**<ConnectionFailureReason>

Gets desktops with a specific reason for the last recorded connection failure. This value is None if the last connection was successful or if there has been no attempt to connect to the desktop yet.

Valid values are None, SessionPreparation, RegistrationTimeout, ConnectionTimeout, Licensing, Ticketing, and Other.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastConnectionTime**<DateTime>

Gets desktops that last connected at a specific time. This is the time that the broker detected that the connection attempt either succeeded or failed.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastConnectionUser**<String>

Gets desktops where a specific user name last attempted a connection (in the form 'domain\user').

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastDeregistrationReason**<DeregistrationReason>

Gets desktops whose broker last recorded a specific deregistration reason.

Valid values are \$null, AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastDeregistrationTime**<DateTime>

Gets desktops that were last deregistered by a specific time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastErrorReason**<String>

Gets desktops with the specified last error reason.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastErrorTime**<DateTime>

Gets desktops with the specified last error time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastHostingUpdateTime**<DateTime>

Gets desktops with a specific time that the hosting information was last updated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LaunchedViaHostName**<String>

Gets desktops with a specific host name of the StoreFront server from which the user launched the session.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LaunchedViaIP**<String>

Gets desktops with a specific IP address of the StoreFront server from which the user launched the session.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineInternalState**<MachineInternalState>

Gets desktops with the specified internal machine state.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineUid**<Int32>

Gets desktops with a specific machine UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSType**<String>

Gets desktops by the type of operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSVersion**<String>

Gets desktops by the version of the operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PersistUserChanges**<PersistUserChanges>

Gets desktops by the location where the user changes are persisted.

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the Pvd.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PowerActionPending**<Boolean>

Gets desktops with a specific power action pending state.

Valid values are \$true or \$false.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PowerState**<PowerState>

Gets desktops with a specific power state.

Valid values are Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Protocol**<String>

Gets desktops with connections using a specific protocol, for example HDX, RDP, or Console.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ProvisioningType**<ProvisioningType>

Specifies the provisioning type for the catalog. Values can be:

- o Manual - No provisioning.
- o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).
- o MCS - Machine provisioned by MCS (machine must be VM).

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedApplication**<String>

Gets desktops with a specific application published on them (identified by its browser name).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName**<String>

Gets desktops with a specific published name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvdStage**<PvdStage>

Gets machines with a specific personal vDisk stage.

Valid values are None, Requested, Starting, Working and Failed.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RegistrationState**<RegistrationState>

Gets desktops with a specific registration state.

Valid values are Unregistered, Initializing, Registered and AgentError.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaActive**<Boolean>

Gets desktops depending on whether the current session uses SecureICA or not.

Session properties are always null for multi-session desktops.



|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaRequired**<Boolean>

Gets desktops configured with a particular SecureIcaRequired setting. Note that the desktop setting of \$null indicates that the desktop group value is used.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionHidden**<Boolean>

Gets desktops by whether their sessions are hidden or not. Hidden sessions are treated as though they do not exist when launching sessions; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionId**<Int32>

Deprecated.

Gets desktops by session ID, a unique identifier that Remote Desktop Services uses to track the session but it is only unique on that machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionState**<SessionState>

Gets desktops with a specific session state.

Valid values are \$null, Other, PreparingSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession, and Unknown.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionStateChangeTime<DateTime>**

Gets desktops whose sessions last changed state at a specific time.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUid<Int64>**

Gets desktops with a specific session UID (\$null for no session).

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUserName<String>**

Gets desktops with a specific user name for the current session (in the form 'domain\user').

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUserSID<String>**

Gets desktops with a specific SID of the current session user.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SID<String>**

Gets desktops with a specific machine SID.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-SmartAccessTag<String>**

Gets desktops where the session has the specific SmartAccess tag.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartTime<DateTime>**

Gets desktops with a specific session start time.

Session properties are always null for multi-session desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SummaryState<DesktopSummaryState>**

Gets desktops with a specific summary state.

Valid values are Off, Unregistered, Available, Disconnected, and InUse.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Tag<String>**

Gets desktops with a specific tag.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WillShutdownAfterUse<Boolean>**

Gets desktops depending on whether they shut down after use or not.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationUid**<Int32>

Gets desktops with a specific published application (identified by its UID).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|               |                                                         |
|---------------|---------------------------------------------------------|
| Required?     | false                                                   |
| Default Value | The default sort order is by name or unique identifier. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.BrokerAdmin.SDK.GroupInfo

Each GroupInfo object represents one group, and contains the following properties:

-- Count: The count of desktops in this group.

-- Name: The value of the property the desktops were grouped by (as a string).

If you do not specify -SortBy, groups are sorted with the largest count first.

**Notes**

To compare dates or times, use -Filter and relative comparisons. For more information, see about\_Broker\_Filtering and the examples.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Group-BrokerDesktop -Property SummaryState -DesktopGroupName dg1
Group desktops from the dg1 group by summary state.
```

----- **EXAMPLE 2** -----

```
C:\PS> Group-BrokerDesktop -Property LastConnectionFailure -Filter { LastConnectionFailure -ne "None" -and LastConnectionTime -ge '-7' } -MaxRecordCount 1
For desktops where the last connection attempt failed, list the most common reason for failure, ignoring connections that failed over a week ago.
```

----- **EXAMPLE 3** -----

```
C:\PS> Group-BrokerDesktop -Property HostingServerName -DesktopCondition ICALatency -SortBy Name
List alphabetically the hypervisor servers hosting desktops that are currently experiencing high network latency.
```

# Group-BrokerMachine

Jan 04, 2017

Groups and counts machines with the same value for a specified property.

## Syntax

```
Group-BrokerMachine [-UId <Int32> -Property <String> [-AdminAddress <String>] [<CommonParameters>]
```

```
Group-BrokerMachine -Property <String> [[-MachineName <String>] [-AgentVersion <String>] [-AllocationType <AllocationType>] [-ApplicationInUse <String>] [-AssignedClientName <String>] [-AssignedIPAddress <String>] [-AssociatedUserFullName <String>] [-AssociatedUserName <String>] [-AssociatedUserSID <String>] [-AssociatedUserUPN <String>] [-BrowserName <String>] [-CatalogName <String>] [-CatalogUId <Int32>] [-CatalogUUID <Guid>] [-ColorDepth <ColorDepth>] [-ControllerDNSName <String>] [-DeliveryType <DeliveryType>] [-Description <String>] [-DesktopCondition <String>] [-DesktopGroupName <String>] [-DesktopGroupUId <Int32>] [-DesktopGroupUUID <Guid>] [-DesktopKind <DesktopKind>] [-DesktopUId <Int32>] [-DNSName <String>] [-FaultState <MachineFaultState>] [-FunctionalLevel <FunctionalLevel>] [-HostedMachineId <String>] [-HostedMachineName <String>] [-HostingServerName <String>] [-HypervisorConnectionName <String>] [-HypervisorConnectionUId <Int32>] [-HypHypervisorConnectionUId <Guid>] [-IconUId <Int32>] [-ImageOutOfDate <Boolean>] [-InMaintenanceMode <Boolean>] [-IPAddress <String>] [-IsAssigned <Boolean>] [-IsPhysical <Boolean>] [-LastConnectionFailure <ConnectionFailureReason>] [-LastConnectionTime <DateTime>] [-LastConnectionUser <String>] [-LastDeregistrationReason <DeregistrationReason>] [-LastDeregistrationTime <DateTime>] [-LastErrorReason <String>] [-LastErrorTime <DateTime>] [-LastHostingUpdateTime <DateTime>] [-LastPvdErrorReason <String>] [-LastPvdErrorTime <DateTime>] [-LoadIndex <Int32>] [-MachineInternalState <MachineInternalState>] [-Metadata <String>] [-OSType <String>] [-OSVersion <String>] [-PersistUserChanges <PersistUserChanges>] [-PowerActionPending <Boolean>] [-PowerState <PowerState>] [-ProvisioningType <ProvisioningType>] [-PublishedApplication <String>] [-PublishedName <String>] [-PvdEstimatedCompletionTime <DateTime>] [-PvdPercentDone <Int32>] [-PvdStage <PvdStage>] [-PvdUpdateStartTime <DateTime>] [-RegistrationState <RegistrationState>] [-ScheduledReboot <ScheduledReboot>] [-SecureIcaRequired <Boolean>] [-SessionAutonomouslyBrokered <Boolean>] [-SessionClientAddress <String>] [-SessionClientName <String>] [-SessionClientVersion <String>] [-SessionConnectedViaHostName <String>] [-SessionConnectedViaIP <String>] [-SessionCount <Int32>] [-SessionDeviceId <String>] [-SessionHardwareId <String>] [-SessionHidden <Boolean>] [-SessionKey <Guid>] [-SessionLaunchedViaHostName <String>] [-SessionLaunchedViaIP <String>] [-SessionProtocol <String>] [-SessionSecureIcaActive <Boolean>] [-SessionsEstablished <Int32>] [-SessionSmartAccessTag <String>] [-SessionsPending <Int32>] [-SessionStartTime <DateTime>] [-SessionState <SessionState>] [-SessionStateChangeTime <DateTime>] [-SessionSupport <SessionSupport>] [-SessionType <SessionType>] [-SessionUId <Int64>] [-SessionUserName <String>] [-SessionUserSID <String>] [-SID <String>] [-SummaryState <DesktopSummaryState>] [-SupportedPowerActions <String[]>] [-Tag <String>] [-UUID <Guid>] [-VMToolsState <VMToolsState>] [-WillShutdownAfterUse <Boolean>] [-WindowsConnectionSetting <WindowsConnectionSetting>] [-AssignedUserSID <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Filters machines using the specified criteria, then groups and counts matching machines with the same value for a particular property. The number of machines in the group, and the property value for the group, is output. For example:

```
C:\PS> Group-BrokerMachine -Property SummaryState
```

```
Count Name
```

```
-----
```

```
43 Available
```

```
17 InUse
```

```
3 Disconnected
```

Filtering supports the same options as the Get-BrokerMachine cmdlet, and allows filtering on both machine and session properties.

Group-BrokerMachine is similar to the standard PowerShell Group-Object, but is faster than piping the output of Get-BrokerMachine into Group-Object when working with many machines.

Note that the MaxRecordCount, ReturnTotalRecordCount, Skip, and SortBy parameters apply to GroupInfo records output rather than the filtered machines.

## Related topics

[Get-BrokerMachine](#)

[Group-Object](#)

[Parameters](#)

**-UId**<Int32>

Gets a machine with a specific UID.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String>

Selects the property by which matching machines are grouped.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName**<String>

Gets machines with a specific machine name (in the form domain\machine).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AgentVersion**<String>

Gets machines with a specific Virtual Delivery Agent version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AllocationType**<AllocationType>

Gets machines from catalogs with the specified allocation type.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationInUse**<String>

Gets machines running a specified published application. String comparisons are case-insensitive.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedClientName**<String>

Gets machines that have been assigned to the specific client name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedIPAddress**<String>

Gets machines that have been assigned to the specific client IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserFullName**<String>

Gets machines with an associated user identified by their full name (usually 'first-name last-name').

Associated users are all current users of a desktop, plus the assigned users for private desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserName**<String>

Gets machines with an associated user identified by their user name (in the form 'domain\user').

Associated users are all current users of a desktop, plus the assigned users for private desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserSID**<String>

Gets machines with an associated user identified by their Windows SID.

Associated users are all current users of a desktop, plus the assigned users for private desktops.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssociatedUserUPN<String>**

Gets machines with an associated user identified by their User Principle Name (in the form 'user@domain').

Associated users are all current users of a desktop, plus the assigned users for private desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrowserName<String>**

Gets assigned machines backing desktop resources that have browser names matching the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CatalogName<String>**

Gets machines from the catalog with the specific name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CatalogUid<Int32>**

Gets machines from the catalog with the specific UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CatalogUUID<Guid>**

Gets machines from the catalog with the specific UUID.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ColorDepth**<ColorDepth>

Gets machines configured with a specific color depth.

Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ControllerDNSName**<String>

Gets machines by the DNS name of the controller they are registered with.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DeliveryType**<DeliveryType>

Gets machines of a particular delivery type.

Valid values are AppsOnly, DesktopsOnly, DesktopsAndApps

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Get machines by description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopCondition**<String>

Gets machines with an outstanding desktop condition.

Valid values are:

- o CPU: Indicates the machine has high CPU usage
- o ICALatency: Indicates the network latency is high

o UPMLogonTime: Indicates that the profile load time was high

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupName**<String>

Gets machines from a desktop group with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Gets machines from a desktop group with a specific UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUUID**<Guid>

Gets machines from a desktop group with a specific UUID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopKind**<DesktopKind>

Deprecated: Use AllocationType parameter.

Gets machines of a particular kind.

Valid values are Private, Shared.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopUid**<Int32>

Gets the machine that corresponds to the desktop with the specific UID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DNSName**<String>

Gets machines with the specific DNS name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-FaultState**<MachineFaultState>

Gets machines currently in the specified fault state.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-FunctionalLevel**<FunctionalLevel>

Gets machines with a specific FunctionalLevel.

Valid values are L5, L7, L7\_6

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineId**<String>

Gets machines with the specific machine ID known to the hypervisor.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineName**<String>

Gets machines with the specific machine name known to the hypervisor.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostingServerName<String>**

Gets machines by the name of the hosting hypervisor server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionName<String>**

Gets machines with the specific name of the hypervisor connection hosting them.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionUid<Int32>**

Gets machines with the specific UID of the hypervisor connection hosting them.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypHypervisorConnectionUid<Guid>**

Gets machines with the specific UUID of the hypervisor connection hosting them.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconUid<Int32>**

Gets machines by configured icon. Note that machines with a null IconUid use the icon of the desktop group.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ImageOutOfDate**<Boolean>

Gets machines depending on whether their disk image is out of date or not (for machines provisioned using MCS only).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InMaintenanceMode**<Boolean>

Gets machines by whether they are in maintenance mode or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IPAddress**<String>

Gets machines with a specific IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsAssigned**<Boolean>

Gets machines according to whether they are assigned or not. Machines may be assigned to one or more users or groups, a client IP address or a client endpoint name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsPhysical**<Boolean>

Gets machines according to whether they can be power managed by XenDesktop or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastConnectionFailure**<ConnectionFailureReason>

Gets machines with a specific reason for the last recorded connection failure. This value is None if the last connection was successful or if there has been no attempt to connect to the machine yet.

Valid values are None, SessionPreparation, RegistrationTimeout, ConnectionTimeout, Licensing, Ticketing, and Other.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastConnectionTime**<DateTime>

Gets machines to which a user session connection occurred at a specific time. This is the time that the broker detected that the connection attempt either succeeded or failed.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastConnectionUser**<String>

Gets machines where a specific user name last attempted a connection (in the form 'domain\user').

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastDeregistrationReason**<DeregistrationReason>

Gets machines whose broker last recorded a specific deregistration reason.

Valid values are Null, AgentShutdown, AgentSuspended, AgentRequested, IncompatibleVersion, AgentAddressResolutionFailed, AgentNotContactable, AgentWrongActiveDirectoryOU, EmptyRegistrationRequest, MissingRegistrationCapabilities, MissingAgentVersion, InconsistentRegistrationCapabilities, NotLicensedForFeature, UnsupportedCredentialSecurityVersion, InvalidRegistrationRequest, SingleMultiSessionMismatch, FunctionalLevelTooLowForCatalog, FunctionalLevelTooLowForDesktopGroup, PowerOff, DesktopRestart, DesktopRemoved, AgentRejectedSettingsUpdate, SendSettingsFailure, SessionAuditFailure, SessionPrepareFailure, ContactLost, SettingsCreationFailure, UnknownError and BrokerRegistrationLimitReached.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LastDeregistrationTime**<DateTime>

Gets machines by the time that they were last deregistered.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Last ErrorReason<String>**

Gets machines with the specified last error reason.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Last ErrorTime<DateTime>**

Gets machines with the specified last error time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Last HostingUpdateTime<DateTime>**

Gets machines with a specific time that the hosting information was last updated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Last PvdErrorReason<String>**

Gets machines with the specified last Personal vDisk preparation error reason.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Last PvdErrorTime<DateTime>**

Gets machines with the specified last Personal vDisk preparation error time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-LoadIndex<Int32>**

Gets machines by their current load index.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineInternalState<MachineInternalState>**

Gets machines with the specified internal state.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata<String>**

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSType<String>**

Gets machines by the type of operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSVersion<String>**

Gets machines by the version of the operating system they are running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PersistUserChanges**<PersistUserChanges>

Gets machines according to the location where user changes are persisted. Values can be:

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the Pvd.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PowerActionPending**<Boolean>

Gets machines depending on whether a power action is pending or not.

Valid values are \$true or \$false.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PowerState**<PowerState>

Gets machines with a specific power state.

Valid values are Unmanaged, Unknown, Unavailable, Off, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ProvisioningType**<ProvisioningType>

Specifies the provisioning type for the catalog. Values can be:

- o Manual - No provisioning.
- o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).
- o MCS - Machine provisioned by MCS (machine must be VM).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedApplication**<String>

Gets machines with a specific application published to them (identified by its browser name).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName**<String>

Gets desktops with a specific published name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvdEstimatedCompletionTime**<DateTime>

If preparation of the Personal vDisk is currently in progress for this machine, this reports an estimation of the time at which the process will be complete.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvdPercentDone**<Int32>

Gets machines a specific percentage through the Personal vDisk preparation process.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvdStage**<PvdStage>

Gets machines at a specific personal vDisk stage.

Valid values are None, Requested, Starting, Working and Failed.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvdUpdateStartTime**<DateTime>

If preparation of the Personal vDisk is currently in progress for this machine, this reports when the update process began.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RegistrationState**<RegistrationState>

Gets machines in a specific registration state.

Valid values are Unregistered, Initializing, Registered, and AgentError.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ScheduledReboot**<ScheduledReboot>

Gets machines according to their current status with respect to any scheduled reboots (for either scheduled desktop group reboots or image rollout purposes). Valid values are:

- o None - No reboot currently scheduled.
- o Pending - Reboot scheduled but machine still available for use.
- o Draining - Reboot scheduled. New logons are disabled, but reconnections to existing sessions are allowed.
- o InProgress - Machine is actively being rebooted.
- o Natural - Natural reboot in progress. Machine is awaiting a restart.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaRequired**<Boolean>

Gets machines configured with a particular SecureIcaRequired setting. Note that the machine setting of \$null indicates that the desktop group value is used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionAutonomouslyBrokered**<Boolean>

Gets machines according to whether their current session is autonomously brokered or not. Autonomously brokered sessions are HDX sessions established by direct connection without being brokered.

Session properties are always null for multi-session machines.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionClientAddress**<String>

Gets machines with a specific client IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionClientName**<String>

Gets machines with a specific client name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionClientVersion**<String>

Gets machines with a specific client version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionConnectedViaHostName**<String>

Gets machines with a specific host name of the incoming connection. This is usually a proxy or Citrix Access Gateway server.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionConnectedViaIP**<String>

Gets machines with a specific IP address of the incoming connection.

Session properties are always null for multi-session machines.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionCount**<Int32>

Gets machines according to the total number of both pending and established user sessions on the machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionDeviceId**<String>

Gets machines with a specific client device ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionHardwareId**<String>

Gets machines with a specific client hardware ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionHidden**<Boolean>

Gets machines by whether their sessions are hidden or not. Hidden sessions are treated as though they do not exist when launching sessions using XenDesktop; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionKey**<Guid>

Gets machine running the session with the specified unique key.

Session properties are always null for multi-session machines.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-SessionLaunchedViaHostName<String>**

Gets machines with a specific host name of the Web Interface server from which the user launched the session.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionLaunchedViaIP<String>**

Gets machines with a specific IP address of the Web Interface server from which the user launched the session.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionProtocol<String>**

Gets machines with connections using a specific protocol, for example HDX, RDP, or Console.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionSecureIcaActive<Boolean>**

Gets machines depending on whether the current session uses SecureICA or not.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionsEstablished<Int32>**

Gets machines according to the number of established user sessions present on the machine.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionSmartAccessTag**<String>

Gets session machines where the session has the specific SmartAccess tag.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionsPending**<Int32>

Get machines according to the number of pending user sessions for the machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionStartTime**<DateTime>

Gets machines with a specific session start time.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionState**<SessionState>

Gets machines with a specific session state.

Valid values are \$null, Other, PreparingSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession, and Unknown.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionStateChangeTime**<DateTime>

Gets machines whose sessions last changed state at a specific time.

Session properties are always null for multi-session machines.



|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionSupport**<SessionSupport>

Gets machines that have the specified session capability. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionType**<SessionType>

Gets machines with a specific session state.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUid**<Int64>

Gets single-session machines with a specific session UID (\$null for no session).

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUserName**<String>

Gets machines with a specific user name for the current session (in the form 'domain\user').

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionUserSID**<String>

Gets machines with a specific SID of the current session user.

Session properties are always null for multi-session machines.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SID**<String>

Gets machines with a specific machine SID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SummaryState**<DesktopSummaryState>

Gets machines with a specific summary state.

Valid values are Off, Unregistered, Available, Disconnected, and InUse.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SupportedPowerActions**<String[]>

A list of power actions supported by this machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Tag**<String>

Gets machines where the session has the given SmartAccess tag.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

---

**-UUID**<Guid>

Gets machines with the specified value of UUID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-VMToolsState**<VMToolsState>

Gets machines with a specific VM tools state.

Valid values are NotPresent, Unknown, NotStarted, and Running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WillShutdownAfterUse**<Boolean>

Gets machines depending on whether they shut down after use or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WindowsConnectionSetting**<WindowsConnectionSetting>

Gets machines according to their current Windows connection setting (logon mode). Valid values are:

- o LogonEnabled - All logons are enabled.
- o Draining - New logons are disabled, but reconnections to existing sessions are allowed.
- o DrainingUntilRestart - Same as Draining, but setting reverts to LogonEnabled when machine next restarts.
- o LogonDisabled - All logons and reconnections are disabled.

This is a Windows setting and is not controlled by XenDesktop. It applies only to multi-session machines; for single-session machines its value is always LogonEnabled.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedUserSID**<String>

Gets machines with the specific SID of the user to whom the desktop is assigned.

---

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Broker\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by `-ReturnTotalRecordCount`.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-Sort By**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See [about\\_Broker\\_Filtering](#) for details.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.GroupInfo

Each GroupInfo object represents one group, and contains the following properties:

-- Count: The count of machines in this group.

-- Name: The value of the property the machines were grouped by (as a string).

If you do not specify -SortBy, groups are sorted with the largest count first.

**Notes**

To compare dates or times, use -Filter and relative comparisons. For more information, see about\_Broker\_Filtering and the examples.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Group-BrokerMachine -Property SummaryState -DesktopGroupName dg1
Group machines from the dg1 group by summary state.
```

----- **EXAMPLE 2** -----

```
C:\PS> Group-BrokerMachine -Property LastConnectionFailure -Filter { LastConnectionFailure -ne "None" -and LastConnectionTime -ge '-7' } -MaxRecordCount 1
For machines where the last connection attempt failed, list the most common reason for failure, ignoring connections that failed over a week ago.
```

----- **EXAMPLE 3** -----

```
C:\PS> Group-BrokerMachine -Property HostingServerName -DesktopCondition ICALatency -SortBy Name
List alphabetically the hypervisor servers hosting machines that are currently experiencing high network latency.
```

# Group-BrokerSession

Jan 04, 2017

Groups and counts sessions with the same value for a specified property.

## Syntax

```
Group-BrokerSession [-Uid] <Int64> -Property <String> [-AdminAddress <String>] [<CommonParameters>]
```

```
Group-BrokerSession -Property <String> [[-SessionKey] <Guid>] [-AgentVersion <String>] [-ApplicationInUse <String>] [-AppState <SessionAppState>] [-AppStateLastChangeTime <DateTime>] [-AutonomouslyBrokered <Boolean>] [-BrokeringDuration <Int32>] [-BrokeringTime <DateTime>] [-BrokeringUserName <String>] [-BrokeringUserSID <String>] [-CatalogName <String>] [-ClientAddress <String>] [-ClientName <String>] [-ClientPlatform <String>] [-ClientProductId <Int32>] [-ClientVersion <String>] [-ConnectedViaHostName <String>] [-ConnectedViaIP <String>] [-ConnectionMode <ConnectionMode>] [-ControllerDNSName <String>] [-DesktopGroupName <String>] [-DesktopGroupUid <Int32>] [-DesktopKind <DesktopKind>] [-DesktopSID <String>] [-DesktopUid <Int32>] [-DeviceId <String>] [-DNSName <String>] [-EstablishmentDuration <Int32>] [-EstablishmentTime <DateTime>] [-HardwareId <String>] [-Hidden <Boolean>] [-HostedMachineName <String>] [-HostingServerName <String>] [-HypervisorConnectionName <String>] [-ImageOutOfDate <Boolean>] [-InMaintenanceMode <Boolean>] [-IPAddress <String>] [-IsAnonymousUser <Boolean>] [-IsPhysical <Boolean>] [-LaunchedViaHostName <String>] [-LaunchedViaIP <String>] [-LogoffInProgress <Boolean>] [-LogonInProgress <Boolean>] [-MachineName <String>] [-MachineSummaryState <DesktopSummaryState>] [-MachineUid <Int32>] [-Metadata <String>] [-OSType <String>] [-PersistUserChanges <PersistUserChanges>] [-PowerState <PowerState>] [-Protocol <String>] [-ProvisioningType <ProvisioningType>] [-ReceiverIPAddress <String>] [-ReceiverName <String>] [-SecureIcaActive <Boolean>] [-SessionId <Int32>] [-SessionState <SessionState>] [-SessionStateChangeTime <DateTime>] [-SessionSupport <SessionSupport>] [-SessionType <SessionType>] [-StartTime <DateTime>] [-UntrustedUserName <String>] [-UserFullName <String>] [-UserName <String>] [-UserSID <String>] [-UserUPN <String>] [-ApplicationUid <Int32>] [-SharedDesktopUid <Int32>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Filters sessions using the specified criteria, then groups and counts matching sessions with the same value for a particular property. The number of sessions in the group, and the property value for the group, is output. For example:

```
C:\PS> Group-BrokerSession -Property SessionState
```

```
Count Name
```

```
-----
```

```
43 Active
```

```
17 NonBrokeredSession
```

```
3 Disconnected
```

Filtering supports the same options as the Get-BrokerSession cmdlet, and allows filtering on both machine and session properties.

Group-BrokerSession is similar to the standard PowerShell Group-Object, but is faster than piping the output of Get-

BrokerSession into Group-Object when working with many machines.

Note that the MaxRecordCount, ReturnTotalRecordCount, Skip, and SortBy parameters apply to GroupInfo records output rather than the filtered sessions.

## Related topics

[Get-BrokerSession](#)

Group-Object

Parameters

**-Uid**<Int64>

Get session by its Uid.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String>

Selects the property by which matching sessions are grouped.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionKey**<Guid>

Gets session having the specified unique key.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AgentVersion**<String>

Gets sessions with a specific Virtual Desktop Agent version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationInUse**<String>

Gets sessions running specific applications (identified by their SDK Name property).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AppState**<SessionAppState>

Get sessions by their app state.

Valid values are PreLogon, PreLaunched, Active, Desktop, Linger and NoApps.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AppStateLastChangeTime**<DateTime>

Get sessions by their app state change time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AutonomouslyBrokered**<Boolean>



Gets sessions according to whether they are autonomously brokered or not. Autonomously brokered sessions are HDX sessions established by direct connection without being brokered.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrokeringDuration**<Int32>

Gets session with a specific time taken to broker. In general, Citrix recommends using -Filter and relative comparisons.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrokeringTime**<DateTime>

Get sessions brokered at a specific time. In general, Citrix recommends using -Filter and relative comparisons.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrokeringUserName**<String>

Get sessions by brokering user.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-BrokeringUserSID**<String>

Get sessions by brokering user SID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CatalogName**<String>

Gets sessions on machines from a specific catalog name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ClientAddress**<String>

Get sessions by client IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Client Name**<String>

Get sessions by client name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Client Platform**<String>

Get sessions by client platform.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Client Product Id**<Int32>

Get sessions by client product ID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Client Version**<String>

Get sessions by client version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ConnectedViaHostName**<String>

Get sessions by host name of the incoming connection. This is usually a proxy or Citrix Access Gateway server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ConnectedViaIP**<String>

Get sessions by IP address of the incoming connection.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ConnectionMode**<ConnectionMode>

Gets sessions by the way in which the most recent connection to the session was established.

Valid modes are Brokered, Unbrokered, LeasedConnection, VdaHighAvailabilityMode, ThirdPartyBroker, and ThirdPartyBrokerWithLicensing.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ControllerDNSName**<String>

Gets sessions that are hosted on machines which are registered with a specific controller.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupName**<String>

Gets sessions from a desktop group with the specified name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupUid**<Int32>

Gets sessions from a desktop group with the specified UID.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopKind**<DesktopKind>

Gets sessions on a desktop of a particular kind.

Valid values are Private and Shared.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopSID**<String>

Get sessions by desktop SID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopUid**<Int32>

Get sessions by desktop Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DeviceId**<String>

Get sessions by client device id.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DNSName<String>**

Gets sessions by their machine's DNS name.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-EstablishmentDuration<Int32>**

Gets sessions which took a specific time to establish. In general, Citrix recommends using -Filter and relative comparisons.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-EstablishmentTime<DateTime>**

Gets sessions which became established at a particular time. In general, Citrix recommends using -Filter and relative comparisons.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HardwareId<String>**

Get sessions by client hardware id.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Hidden**<Boolean>

Get sessions by whether they are hidden or not. Hidden sessions are treated as though they do not exist when brokering sessions; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineName**<String>

Gets sessions by their machine's name as known to its hypervisor.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostingServerName**<String>

Gets sessions hosted by a machine with a specific name of the hosting hypervisor server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorConnectionName**<String>

Gets sessions hosted by a machine with a specific name of the hosting hypervisor connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Accept Pipeline Input? |       |

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ImageOutOfDate**<Boolean>

Gets sessions hosted by a machine with a specific ImageOutOfDate setting.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InMaintenanceMode**<Boolean>

Gets sessions hosted by a machine with a specific InMaintenanceMode setting.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IPAddress**<String>

Gets sessions hosted by a machine with a specific IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsAnonymousUser**<Boolean>

Gets sessions depending on whether they were established anonymously (\$true) or not (\$false). An anonymous session is established without user credentials and a temporary local user account is used.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |



|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-IsPhysical**<Boolean>

Gets sessions hosted on machines where the flag indicating if the machine can be power managed by the Citrix Broker Service matches the requested value. Where the power state of the machine cannot be controlled, specify \$true, otherwise \$false.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LaunchedViaHostName**<String>

Get sessions by the host name of the Web Interface server from which a user launches a session.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LaunchedViaIP**<String>

Get sessions by the IP address of the Web Interface server from which a user launches a session.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LogoffInProgress**<Boolean>

Gets sessions by whether they are in the process of being logged off or not.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-LogonInProgress**<Boolean>

Gets sessions by whether they are still executing user logon processing or not.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineName**<String>

Gets sessions by their machine name (in the form DOMAIN\machine).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineSummaryState**<DesktopSummaryState>

Gets sessions on a machine with a specific summary state.

Valid values are Off, Unregistered, Available, Disconnected, Preparing, and InUse.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MachineUid**<Int32>

Gets sessions on a machine with the specified UID.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OSType**<String>

Gets sessions with a specific type of operating system.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PersistUserChanges**<PersistUserChanges>

Gets sessions where the user changes are persisted in a particular manner. Values can be:

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the Pvd.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PowerState**<PowerState>

Gets sessions on machines in the specified power state.

Valid values are Unmanaged, Unknown, Unavailable, On, Suspended, TurningOn, TurningOff, Suspending, and Resuming.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Protocol**<String>

Get sessions by connection protocol. Valid values are HDX, RDP and Console.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ProvisioningType**<ProvisioningType>

Gets sessions hosted on machines provisioned in a particular manner. Values can be:

- o Manual - No automated provisioning.
- o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).
- o MCS - Machine provisioned by MCS (machine must be VM).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReceiverIPAddress**<String>

Gets sessions with the specified client IP address supplied by Receiver (for example, StoreFront) when the session was launched, or reconnected.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ReceiverName**<String>

Gets sessions with the specified client name supplied by Receiver (for example, StoreFront) when the session was launched, or reconnected.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaActive**<Boolean>

Get sessions by their use of SecureICA.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionId**<Int32>

Deprecated.

Gets sessions by session ID, a unique identifier that Remote Desktop Services uses to track the session but it is only unique on that machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionState**<SessionState>

Get sessions by their state.

Valid values are Other, PreparingNewSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession, and Unknown.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionStateChangeTime**<DateTime>

Get sessions by their last state change time. In general, Citrix recommends using -Filter and relative comparisons.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionSupport**<SessionSupport>

Gets sessions hosted on machines which support the required pattern of sessions. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionType**<SessionType>

Get sessions by their type.

Valid values are Application and Desktop.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartTime**<DateTime>

Get sessions by their start time. In general, Citrix recommends using -Filter and relative comparisons.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UntrustedUserName**<String>

Gets sessions by the untrusted user name reported directly from the machine (in the form DOMAIN\user).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserFullName**<String>

Gets sessions by user's full name (usually 'first-name last-name').

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserName**<String>

Get sessions by user name (in the form DOMAIN\user).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserSID**<String>

Get sessions by user's Windows SID.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserUPN**<String>

Gets sessions by user's User Principal Name (in the form user@domain).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ApplicationUid**<Int32>

Get sessions running the application with the specified Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SharedDesktopUid**<Int32>

Get sessions by SharedDesktop Uid.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, this causes the cmdlet to output an error record containing the number of records available. This error



record is additional information and does not affect the objects written to the output pipeline. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell style filter expression. See about\_Broker\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.GroupInfo

Each GroupInfo object represents one group, and contains the following properties:

- Count: The count of sessions in this group.
- Name: The value of the property the sessions were grouped by (as a string).

If you do not specify -SortBy, groups are sorted with the largest count first.

**Notes**

To compare dates or times, use -Filter and relative comparisons. For more information, see about\_Broker\_Filtering and the examples.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Group-BrokerSession -Property SessionState -DesktopGroupName dg1  
Group sessions on machines from the dg1 group by session state.

----- **EXAMPLE 2** -----

```
C:\PS> Group-BrokerSession -Property ClientName -ClientName 'ThinClient*' -SortBy Name
```

List alphabetically the names of the clients connected to the site, but only show clients whose names starts with 'ThinClient'.

# Import-BrokerDesktopPolicy

Jan 04, 2017

Sets the site wide Citrix Group Policy settings for the site.

## Syntax

```
Import-BrokerDesktopPolicy [-Policy] <Byte[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Import-BrokerDesktopPolicy sets the site wide Citrix Group Policy settings. A successful call to this cmdlet will result in the supplied data being uploaded to every machine in the site prior to its next session launch.

## Related topics

[Export-BrokerDesktopPolicy](#)

[New-BrokerConfigurationSlot](#)

[New-BrokerMachineConfiguration](#)

## Parameters

**-Policy**<Byte[]>

The configuration data containing the Citrix Group Policy settings to apply to every machine in the site.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

System.Byte[] The configuration data as an opaque binary blob.

### Return Values

None

### Notes

Import-BrokerDesktopPolicy performs a specialized operation. Direct usage of it in scripts is discouraged, and could result in data corruption. It is recommended that this operation be performed via the Citrix Studio.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Import-BrokerDesktopPolicy $policyData
```

This command sets the Citrix Group Policy settings in the site. These policy settings are then applied to every machine prior to the next session launch.

# Move-BrokerAdminFolder

Jan 04, 2017

Moves a folder to another place in the hierarchy, optionally renaming it

## Syntax

```
Move-BrokerAdminFolder [-InputObject] <AdminFolder[]> [-Destination] <AdminFolder> [-NewName <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Move-BrokerAdminFolder [-Name] <String> [-Destination] <AdminFolder> [-NewName <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Move-BrokerAdminFolder cmdlet moves a folder for organising objects for administration purposes (for example, Applications) to another position in the hierarchy.

The following special characters are not allowed in the new FolderName: \ / ; : # . \* ? = < > | [ ] ( ) " ' `

## Related topics

[Get-BrokerAdminFolder](#)

[New-BrokerAdminFolder](#)

## Parameters

**-InputObject**<AdminFolder[]>

The folder(s) to be moved

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

A pattern matching the names of folders to be moved

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Destination**<AdminFolder>

The destination folder the folder being moved should end up in

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-NewName**<String>

The name the new folder should have in the destination folder

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Depends on parameter Parameters can be piped by property name.

### Return Values

None or Citrix.Broker.Admin.SDK.AdminFolder

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AdminFolder object.

### Examples

#### ----- **EXAMPLE 1** -----

```
Move-BrokerAdminFolder F1\XXX\ F2\
```

Moves the folder called XXX within the folder F1\ to a new home in F2\

#### ----- **EXAMPLE 2** -----

```
Move-BrokerAdminFolder F1\XXX\ F2\ -NewName YYY
```

Moves the folder called XXX within the folder F1\ to a new home in F2\ renaming it to YYY in the process



# Move-BrokerApplication

Jan 04, 2017

Move a published application from one admin folder to another

## Syntax

```
Move-BrokerApplication [-InputObject] <Application[]> [-Destination] <AdminFolder> [-NewName <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Move-BrokerApplication [-Name] <String> [-Destination] <AdminFolder> [-NewName <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Move-BrokerApplication cmdlet moves a published application from one place to another in the tree of admin folders, optionally renaming it in the process (if you only want to change the name of the application for administrative purposes and not its location in the tree, use the Rename-BrokerApplication cmdlet).

The location and name of a published application in this sense is only of interest to the administrator, changes do not affect the end-user experience.

## Related topics

[New-BrokerApplication](#)

[Add-BrokerApplication](#)

[Get-BrokerApplication](#)

[Remove-BrokerApplication](#)

[Rename-BrokerApplication](#)

[Set-BrokerApplication](#)

## Parameters

**-InputObject** <Application[]>

The application(s) to be moved

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name** <String>

The application(s) to be moved

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Destination**<AdminFolder>

The destination location within the admin folder hierarchy

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-NewName**<String>

The new name of the application in its new destination

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.Application You can pipe applications to Move-BrokerApplication.

**Return Values**

None or Citrix.Broker.Admin.SDK.Application

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.Application object.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Move-BrokerApplication -Name 'App1' -Destination 'F1\'
Moves the application in the root folder called "App1" to the folder "F1\".
```

----- **EXAMPLE 2** -----

```
C:\PS> Move-BrokerApplication 'F1\App1' 'F2\' -NewName 'Application1'
Moves the application in folder "F1" called "App1" to the folder "F2\", renaming it to "Application1" in the process.
```

# New-BrokerAccessPolicyRule

Jan 04, 2017

Creates a new rule in the site's access policy.

## Syntax

```
New-BrokerAccessPolicyRule [-Name] <String> [-DesktopGroupUid <Int32>] [-AllowedConnections <AllowedConnection>] [-AllowedProtocols <String[]>] [-AllowedUsers <AllowedUser>] [-AllowRestart <Boolean>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedClientIPFilterEnabled <Boolean>] [-ExcludedClientIPs <IPAddressRange[]>] [-ExcludedClientNameFilterEnabled <Boolean>] [-ExcludedClientNames <String[]>] [-ExcludedSmartAccessFilterEnabled <Boolean>] [-ExcludedSmartAccessTags <String[]>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-HdxSslEnabled <Boolean>] [-IncludedClientIPFilterEnabled <Boolean>] [-IncludedClientIPs <IPAddressRange[]>] [-IncludedClientNameFilterEnabled <Boolean>] [-IncludedClientNames <String[]>] [-IncludedSmartAccessFilterEnabled <Boolean>] [-IncludedSmartAccessTags <String[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-BrokerAccessPolicyRule [-Name] <String> [-IncludedDesktopGroups <DesktopGroup[]>] [-IncludedDesktopGroupFilterEnabled <Boolean>] [-AllowedConnections <AllowedConnection>] [-AllowedProtocols <String[]>] [-AllowedUsers <AllowedUser>] [-AllowRestart <Boolean>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedClientIPFilterEnabled <Boolean>] [-ExcludedClientIPs <IPAddressRange[]>] [-ExcludedClientNameFilterEnabled <Boolean>] [-ExcludedClientNames <String[]>] [-ExcludedSmartAccessFilterEnabled <Boolean>] [-ExcludedSmartAccessTags <String[]>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-HdxSslEnabled <Boolean>] [-IncludedClientIPFilterEnabled <Boolean>] [-IncludedClientIPs <IPAddressRange[]>] [-IncludedClientNameFilterEnabled <Boolean>] [-IncludedClientNames <String[]>] [-IncludedSmartAccessFilterEnabled <Boolean>] [-IncludedSmartAccessTags <String[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The `New-BrokerAccessPolicyRule` cmdlet adds a new rule to the site's access policy.

An access policy rule defines a set of connection filters and access control rights relating to a desktop group. These allow fine-grained control of what access is granted to a desktop group based on details of, for example, a user's endpoint device, its address, and the user's identity.

Multiple rules in the access policy can apply to the same desktop group.

For a user to gain access to a desktop group via a rule their connection must match all its enabled include filters, and none of its enabled exclude filters. In addition, for a user to be able to launch a desktop or application resource session from the desktop group, they must have an entitlement to use the resource granted by the entitlement or assignment policies, or by direct machine assignment.

## Related topics

[Get-BrokerAccessPolicyRule](#)

[Set-BrokerAccessPolicyRule](#)

[Rename-BrokerAccessPolicyRule](#)

[Remove-BrokerAccessPolicyRule](#)

## Parameters

**-Name**<String>

Specifies the administrative name of the new rule. Each rule within the site's access policy must have a unique name.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopGroupUid**<Int32>

Specifies the desktop group to which the new rule applies.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedDesktopGroups**<DesktopGroup[]>

This parameter is supported for backward compatibility only. If used only a single desktop group UID can be specified.

The IncludedDesktopGroups and IncludedDesktopGroupFilterEnabled parameters have been superseded by the DesktopGroupUid parameter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-AllowedConnections**<AllowedConnection>

Specifies whether connections must be local or via Access Gateway, and if so whether specified SmartAccess tags must be provided by Access Gateway with the connection. This property forms part of the included SmartAccess tags filter.

Valid values are Filtered, NotViaAG, and ViaAG.

For a detailed description of this property see "help about\_Broker\_AccessPolicy".

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | Filtered              |
| Accept Pipeline Input? | true (ByPropertyName) |

**-AllowedProtocols**<String[]>

Specifies the protocols (for example HDX, RDP) available to the user for sessions delivered from the new rule's desktop group. If the user gains access to a desktop group by multiple rules, the allowed protocol list is the combination of the protocol lists from all those rules.

If the protocol list is empty, access to the desktop group is implicitly denied.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | HDX                   |
| Accept Pipeline Input? | true (ByPropertyName) |

**-AllowedUsers**<AllowedUser>

Specifies the behavior of the included users filter of the new rule. This can restrict access to a list of named users or groups, allow access to any authenticated user, any user (whether authenticated or not), or only non-authenticated users. For a detailed description of this property see "help about\_Broker\_AccessPolicy".

Valid values are Filtered, AnyAuthenticated, Any, AnonymousOnly and FilteredOrAnonymous.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | Filtered              |
| Accept Pipeline Input? | true (ByPropertyName) |

**-AllowRestart**<Boolean>

Specifies if the user can restart sessions delivered from the new rule's desktop group. Session restart is handled as follows: For sessions on single-session power-managed machines, the machine is powered off, and a new session launch request made; for sessions on multi-session machines, a logoff request is issued to the session, and a new session launch request made; otherwise the property is ignored.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | false |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-Description**<String>

Specifies an optional description of the new rule. The text is purely informational for the administrator, it is never visible to the end user.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Enabled**<Boolean>

Specifies whether the new rule is initially enabled. A disabled rule is ignored when evaluating the site's access policy.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedClientIPFilterEnabled**<Boolean>

Specifies whether the excluded client IP address filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedClientIPs**<IPAddressRange[]>

Specifies IP addresses of user devices explicitly denied access to the new rule's desktop group. Addresses can be specified as simple numeric addresses or as subnet masks (for example, 10.40.37.5 or 10.40.0.0/16). This property forms part of the excluded client IP address filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedClientNameFilterEnabled**<Boolean>

Specifies whether the excluded client names filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedClientNames**<String[]>

Specifies names of user devices explicitly denied access to the new rule's desktop group. This property forms part of the excluded client names filter.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedSmartAccessFilterEnabled**<Boolean>

Specifies whether the excluded SmartAccess tags filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedSmartAccessTags**<String[]>

Specifies SmartAccess tags which explicitly deny access to the new rule's desktop group if any occur in those provided by Access Gateway with the user's connection. This property forms part of the excluded SmartAccess tags filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedUserFilterEnabled**<Boolean>

Specifies whether the excluded users filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedUsers**<User[]>

Specifies any users and groups who are explicitly denied access to the new rule's desktop group. This property forms part of the excluded users filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-HdxSslEnabled**<Boolean>

Indicates whether SSL encryption is enabled for sessions delivered from the rule's desktop group.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | \$false               |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedClientIPFilterEnabled**<Boolean>

Specifies whether the included client IP address filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedClientIPs**<IPAddressRange[]>

Specifies IP addresses of user devices allowed access to the new rule's desktop group. Addresses can be specified as simple numeric addresses or as subnet masks (for example, 10.40.37.5 or 10.40.0.0/16). This property forms part of the included client IP address filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedClientNameFilterEnabled**<Boolean>

Specifies whether the included client name filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedClientNames**<String[]>

Specifies names of user devices allowed access to the new rule's desktop group. This property forms part of the included client names filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedSmartAccessFilterEnabled**<Boolean>

Specifies whether the included SmartAccess tags filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedSmartAccessTags**<String[]>

Specifies SmartAccess tags which grant access to the new rule's desktop group if any occur in those provided by Access Gateway with the user's connection. This property forms part of the excluded SmartAccess tags filter.

|               |              |
|---------------|--------------|
| Required?     | false        |
| Default Value | (empty list) |



|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-IncludedUserFilterEnabled**<Boolean>

Specifies whether the included users filter is initially enabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedUsers**<User[]>

Specifies users and groups who are granted access to the new rule's desktop group. This property forms part of the included users filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**-IncludedDesktopGroupFilterEnabled**<Boolean>

This parameter is supported for backward compatibility only. If used the supplied value must be \$true.

The IncludedDesktopGroups and IncludedDesktopGroupFilterEnabled parameters have been superseded by the DesktopGroupUid parameter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**Input Type**

None You cannot pipe input into this cmdlet.

## Return Values

Citrix.Broker.Admin.SDK.AccessPolicyRule

New-BrokerAccessPolicyRule returns the newly created access policy rule.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Tech Support'
```

```
C:\PS> New-BrokerAccessPolicyRule 'UK Tech Support' -IncludedUserFilterEnabled $true -IncludedUsers support\uk-staff -DesktopGroupUid $dg.Uid -AllowedProtocols 'HDX'
```

Creates an access policy rule allowing access to the Tech Support desktop group for all users of the SUPPORT\uk-staff group. Connections to desktop or application resources in the group can only be made using the HDX protocol.

For users to gain access to resources in the group also requires that, depending on the desktop kind of the group, appropriate assignment or entitlement policy rules, or explicit machine assignments exist.

# New-BrokerAdminFolder

Jan 04, 2017

Creates a new admin folder.

## Syntax

```
New-BrokerAdminFolder [-FolderName] <String> [-ParentFolder <AdminFolder>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The New-BrokerAdminFolder cmdlet creates a new folder for organising objects for administration purposes (for example, Applications).

New-BrokerAdminFolder creates the folder object and optionally places it within an existing admin folder if required.

The following special characters are not allowed in the FolderName: \ / ; # . \* ? = < > | [ ] ( ) " ' `

## Related topics

[Get-BrokerAdminFolder](#)

[Remove-BrokerAdminFolder](#)

## Parameters

**-FolderName**<String>

The simple name of the new folder within its parent (if any)

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ParentFolder**<AdminFolder>

The name or UID of the parent folder (if any)

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Depends on parameter Parameters can be piped by property name.

### Return Values

Citrix.Broker.Admin.SDK.AdminFolder

The new admin folder.

### Examples

----- **EXAMPLE 1** -----

New-BrokerAdminFolder F1

Creates an admin folder called F1 under the root folder (i.e. F1\)

----- **EXAMPLE 2** -----

New-BrokerAdminFolder F2 -AdminFolder F1\

Creates an admin folder called F2 under the folder F1\ (i.e. F1\F2\)

# New-BrokerAppAssignmentPolicyRule

Jan 04, 2017

Creates a new application rule in the site's assignment policy.

## Syntax

```
New-BrokerAppAssignmentPolicyRule [-Name] <String> -DesktopGroupUid <Int32> [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The New-BrokerAppAssignmentPolicyRule cmdlet adds a new application rule to the site's assignment policy.

An application rule in the assignment policy defines the users who are entitled to a self-service persistent machine assignment from the rule's desktop group; once assigned the machine can run one or more applications published from the group.

The following constraints apply when creating an application assignment rule for a desktop group:

- o The group's desktop kind must be Private
- o The group's delivery type must be AppsOnly
- o Only a single application rule can apply to a given group
- o Application assignment rules cannot be applied to RemotePC groups.

When a user selects an application published from a private group, a currently unassigned machine is selected from the group and permanently assigned to the user. An application session is then launched to the machine. Subsequent launches are routed directly to the now assigned machine.

Once a machine has been assigned in this way, the original assignment rule plays no further part in access to the machine.

## Related topics

[Get-BrokerAppAssignmentPolicyRule](#)

[Set-BrokerAppAssignmentPolicyRule](#)

[Rename-BrokerAppAssignmentPolicyRule](#)

[Remove-BrokerAppAssignmentPolicyRule](#)

## Parameters

**-Name**<String>

Specifies the administrative name of the new application rule. Each rule in the site's assignment policy must have a unique name (irrespective of whether they are desktop or application rules).

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopGroupUid**<Int32>

Specifies the unique ID of the desktop group to which the new application rule applies.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Description**<String>

Specifies an optional description of the new application rule. The text is purely informational for the administrator, it is never visible to the end user.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Enabled**<Boolean>

Specifies whether the new application rule is initially enabled. A disabled rule is ignored when evaluating the site's assignment policy.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedUserFilterEnabled**<Boolean>

Specifies whether the excluded users filter is initially enabled. If the filter is disabled then any user entries in the filter are ignored when assignment policy rules are evaluated.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedUsers<User[]>**

Specifies the excluded users filter of the new application rule, that is, the users and groups who are explicitly denied an entitlement to a machine assignment from the rule.

This can be used to exclude users or groups who would otherwise gain access by groups specified in the included users filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedUserFilterEnabled<Boolean>**

Specifies whether the included users filter is initially enabled. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to a machine assignment by the new application rule.

Users who would be implicitly granted access when the filter is disabled can still be explicitly denied access using the excluded users filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedUsers<User[]>**

Specifies the included users filter of the new application rule, that is, the users and groups who are granted an entitlement to a machine assignment by the rule.

If a user appears explicitly in the excluded users filter of the rule or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

|           |       |
|-----------|-------|
| Required? | false |
|           |       |

|                        |                       |
|------------------------|-----------------------|
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.AppAssignmentPolicyRule

New-BrokerAppAssignmentPolicyRule returns the newly created application rule in the assignment policy.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Sales Support'
```

```
C:\PS> New-BrokerAppAssignmentPolicyRule 'UK Office' -DesktopGroupUid $dg.Uid -IncludedUsers sales\uk-staff
```

Creates an application rule in the assignment policy that grants all members of the SALES\uk-staff group an entitlement to a single machine from the Sales Support desktop group. The machine can be used for running applications published from the group.



# New-BrokerAppEntitlementPolicyRule

Jan 04, 2017

Creates a new application rule in the site's entitlement policy.

## Syntax

```
New-BrokerAppEntitlementPolicyRule [-Name] <String> -DesktopGroupUid <Int32> [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-SessionReconnection <SessionReconnection>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The New-BrokerAppEntitlementPolicyRule cmdlet adds a new application rule to the site's entitlement policy.

An application rule in the entitlement policy defines the users who are allowed per-session access to a machine to run one or more applications published from the rule's desktop group.

The following constraints apply when creating an application entitlement rule for a desktop group:

- o The group's desktop kind must be Shared
- o The group's delivery type must be AppsOnly or DesktopsAndApps
- o Only a single application rule can apply to a given group

When a user selects an application published from a shared group, a machine is selected from the group on which to run the application. No permanent association exists between the user and the selected machine; once the session ends the association also ends.

Even though only a single application entitlement and therefore session can be defined for a group, the user can still run multiple applications from the group because the applications run within the same session.

## Related topics

[Get-BrokerAppEntitlementPolicyRule](#)

[Set-BrokerAppEntitlementPolicyRule](#)

[Rename-BrokerAppEntitlementPolicyRule](#)

[Remove-BrokerAppEntitlementPolicyRule](#)

## Parameters

**-Name**<String>

Specifies the administrative name of the new application rule. Each rule in the site's entitlement policy must have a unique name (irrespective of whether they are desktop or application rules).

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopGroupUid**<Int32>

Specifies the unique ID of the desktop group to which the new application rule applies.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Description**<String>

Specifies an optional description of the new application rule. The text is purely informational for the administrator, it is never visible to the end user.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Enabled**<Boolean>

Specifies whether the new application rule is initially enabled. A disabled rule is ignored when evaluating the site's entitlement policy.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedUserFilterEnabled**<Boolean>

Specifies whether the excluded users filter is initially enabled. If the filter is disabled then any user entries in the filter are ignored when entitlement policy rules are evaluated.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedUsers**<User[]>

Specifies the excluded users filter of the application rule, that is, the users and groups who are explicitly denied entitlements to published applications from the desktop group.

This can be used to exclude users or groups who would otherwise gain access by groups specified in the included users filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedUserFilterEnabled**<Boolean>

Specifies whether the included users filter is initially enabled. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to an application session by the new rule.

Users who would be implicitly granted access when the filter is disabled can still be explicitly denied access using the excluded users filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedUsers**<User[]>

Specifies the included users filter of the application rule, that is, the users and groups who are granted an entitlement to an application session by the new rule.

If a user appears explicitly in the excluded users filter of the rule or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

|               |              |
|---------------|--------------|
| Required?     | false        |
| Default Value | (empty list) |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-SessionReconnection**<SessionReconnection>

Defines reconnection (roaming) behavior for sessions launched using this rule. Session reconnection control is an experimental and unsupported feature.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | Always                |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule

New-BrokerAppEntitlementPolicyRule returns the newly created application rule in the entitlement policy.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Customer Support'
```

```
C:\PS> New-BrokerAppEntitlementPolicyRule 'UK Office' -DesktopGroupUid $dg.Uid -IncludedUsers support\uk-staff
```

Creates an application rule in the entitlement policy that entitles all members of the SUPPORT\uk-staff group to a machine for running applications published from the Customer Support desktop group.

# New-BrokerApplication

Jan 04, 2017

Creates a new published application.

## Syntax

```
New-BrokerApplication [-Name] <String> -CommandLineExecutable <String> -DesktopGroup <DesktopGroup> [-AdminFolder <AdminFolder>] [-ApplicationType <ApplicationType>] [-BrowserName <String>] [-ClientFolder <String>] [-CommandLineArguments <String>] [-CpuPriorityLevel <CpuPriorityLevel>] [-Description <String>] [-Enabled <Boolean>] [-IconFromClient <Boolean>] [-IconUid <Int32>] [-Priority <Int32>] [-PublishedName <String>] [-SecureCmdLineArgumentsEnabled <Boolean>] [-ShortcutAddedToDesktop <Boolean>] [-ShortcutAddedToStartMenu <Boolean>] [-StartMenuFolder <String>] [-UserFilterEnabled <Boolean>] [-UUID <Guid>] [-Visible <Boolean>] [-WaitForPrinterCreation <Boolean>] [-WorkingDirectory <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The New-BrokerApplication cmdlet creates a new published application in the site.

New-BrokerApplication creates the application object, and associates it with a desktop group. Application objects have three names that identify them (in addition to their Uid): the Name, BrowserName and the PublishedName. The BrowserName is unique across the entire site, and is primarily used internally. The Name is also unique and is what is seen by the administrator; it contains any prefix for an enclosing admin folder (if any). The PublishedName is not unique and is what is seen by the users.

You can create both HostedOnDesktop and InstalledOnClient applications but the ApplicationType cannot be changed later.

The following special characters are not allowed in the Name, BrowserName or the PublishedName properties: \ / ; : # . \* ? = < > | [ ] ( ) ' "

In addition the ` character is not allowed in the Name property.

See about\_Broker\_Applications for more information.

## Related topics

[Add-BrokerApplication](#)

[Remove-BrokerApplication](#)

[Get-BrokerApplication](#)

[Remove-BrokerApplication](#)

[Rename-BrokerApplication](#)

[Move-BrokerApplication](#)

[Set-BrokerApplication](#)

## Parameters

**-Name**<String>

Specifies the name of the application (must be unique within folder).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-CommandLineExecutable**<String>

Specifies the name of the executable file to launch. The full path need not be provided if it's already in the path. Environment variables can also be used.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | (required)            |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopGroup**<DesktopGroup>

Specifies which desktop group this application should be associated with. The association between application and desktop groups can be added or removed using the Add-BrokerApplication and Remove-BrokerApplication cmdlets.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-AdminFolder**<AdminFolder>

The folder in which the new application should reside (if any).

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ApplicationType**<ApplicationType>

Specifies the type of the application: HostedOnDesktop or InstalledOnClient.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (required)            |
| Accept Pipeline Input? | true (ByPropertyName) |

**-BrowserName**<String>

Specifies the internal name for this application. It must be unique in the site.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (same as Name)        |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Client Folder**<String>

Specifies the folder that the application belongs to as the user sees it. This is the application folder that is seen in the Citrix Online Plug-in, in Web Services, and also in the end-user's Start menu. Subdirectories can be specified with '\' character. The following special characters are not allowed: / \* ? < > | " : . Note that this property cannot be set for applications of type InstalledOnClient.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-CommandLineArguments**<String>

Specifies the command-line arguments to use when launching the executable. Environment variables can be used.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-CpuPriorityLevel**<CpuPriorityLevel>

Specifies the CPU priority for the launched process. Valid values are: Low, BelowNormal, Normal, AboveNormal, and High. Note that this property cannot be set for applications of type InstalledOnClient.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | Normal                |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Description**<String>

Specifies the description of the application. This is only seen by Citrix administrators and is not visible to users.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Enabled**<Boolean>

Specifies whether or not this application can be launched.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IconFromClient**<Boolean>

Specifies if the app icon should be retrieved from the application on the client. This is reserved for possible future use, and all applications of type HostedOnDesktop cannot set or change this value.



|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IconUid<Int32>**

Specifies which icon to use for this application. This icon is visible both to the administrator (in the consoles) and to the user. If no icon is specified, then a generic built-in application icon is used.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 2                     |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Priority<Int32>**

Specifies the priority of the mapping between the application and desktop group. A value of zero has the highest priority, with increasing values indicating lower priorities.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PublishedName<String>**

The name seen by end users who have access to this application.

|                        |                                                                  |
|------------------------|------------------------------------------------------------------|
| Required?              | false                                                            |
| Default Value          | The same value as that supplied for the name of the application. |
| Accept Pipeline Input? | true (ByPropertyName)                                            |

**-SecureCmdLineArgumentsEnabled<Boolean>**

Specifies whether the command-line arguments are secured or not. This is reserved for possible future use, and all applications of type HostedOnDesktop can only have this value set to true.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Short cutAddedToDesktop<Boolean>**

Specifies whether or not a shortcut to the application should be placed on the user device. This is valid only for the Citrix Online Plug-in.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ShortcutAddedToStartMenu**<Boolean>

Specifies whether a shortcut to the application should be placed in the user's start menu on their user device.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-StartMenuFolder**<String>

Specifies the name of the start menu folder that holds the application shortcut (if any). This is valid only for the Citrix Online Plug-in. Subdirectories can be specified with '\' character. The following special characters are not allowed: / \* ? < > | " . :

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-UserFilterEnabled**<Boolean>

Specifies whether the application's user filter is enabled or disabled. Where the user filter is enabled, the application is visible only to users who appear in the filter (either explicitly or by virtue of group membership).

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-UUID**<Guid>

An optional GUID for this application.

|                        |                                              |
|------------------------|----------------------------------------------|
| Required?              | false                                        |
| Default Value          | A new GUID is generated if none is supplied. |
| Accept Pipeline Input? | true (ByPropertyName)                        |

**-Visible**<Boolean>

Specifies whether or not this application is visible to users. Note that it's possible for an application to be disabled and still visible.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-WaitForPrinterCreation**<Boolean>

Specifies whether or not the session waits for the printers to be created before allowing the user to interact with the session. Note that this property cannot be set for applications of type InstalledOnClient.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-WorkingDirectory**<String>

Specifies which working directory the executable is launched from. Environment variables can be used.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

Depends on parameter Parameters can be piped by property name.

## Return Values

Citrix.Broker.Admin.SDK.Application

New-BrokerApplication returns an Application object.

## Notes

Usually only the Name is specified with the New-BrokerApplication cmdlet, and the system chooses a BrowserName and PublishedName for you. By default the BrowserName is the same as the Name, if it is unique in the site. If not, then "-x" is appended to the name, where "x" is a number. For instance, if there is already an application with a BrowserName of "Notepad" and a new application is created with a Name of "Notepad", then the new application gets a BrowserName of "Notepad-1". If another "Notepad" is published, it has a BrowserName of "Notepad-2".

That said, the BrowserName can optionally be specified as well.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> New-BrokerApplication -ApplicationType HostedOnDesktop -Name "Notepad" -CommandLineExecutable "notepad.exe" -DesktopGroup PrivateDG1
```

Creates and returns an object for a published application called "Notepad" that launches "notepad.exe".

### ----- EXAMPLE 2 -----

```
C:\PS> $dg = Get-BrokerDesktopGroup "SharedDG1"
C:\PS> $app = New-BrokerApplication -ApplicationType HostedOnDesktop -Name "Notepad" -CommandLineExecutable "notepad.exe" -DesktopGroup $dg
C:\PS> $group = Get-BrokerDesktopGroup -Name "Shared desktop group"
C:\PS> Add-BrokerApplication $app -DesktopGroup $group
C:\PS> $fta = Get-BrokerImportedFTA -ExtensionName ".txt"
C:\PS> New-BrokerConfiguredFTA -ImportedFTA $fta -ApplicationUid $app.Uid
```

This is a much more complete example. It creates an application object to publish Notepad and associates it first with the "SharedDG1" desktop group.

Next it adds an additional desktop group (one that can host applications), and publishes the application to that desktop group. It then gets the ImportedFTA object for the .txt file-type extension (this assumes file-type associations have already been imported), and then configures it so that ".txt" is associated with the published application.

Note: The appropriate access policy and app assignment/entitlement rules must also be configured to allow access to the application.

# New-BrokerAssignmentPolicyRule

Jan 04, 2017

Creates a new desktop rule in the site's assignment policy.

## Syntax

```
New-BrokerAssignmentPolicyRule [-Name] <String> -DesktopGroupUid <Int32> [-ColorDepth <ColorDepth>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IconUid <Int32>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-MaxDesktops <Int32>] [-PublishedName <String>] [-SecureIcaRequired <Boolean>] [-UUID <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The New-BrokerAssignmentPolicyRule cmdlet adds a new desktop rule to the site's assignment policy.

A desktop rule in the assignment policy defines the users who are entitled to self-service persistent machine assignments from the rule's desktop group. A rule defines how many machines a user is allowed from the group for delivery of full desktop sessions.

The following constraints apply when creating a desktop assignment rule for a desktop group:

- o The group's desktop kind must be Private
- o The group's delivery type must be DesktopsOnly
- o Only one desktop assignment rule can be created for RemotePC groups.

When a user selects a machine assignment entitlement from a private group, a currently unassigned machine is selected from the group and permanently assigned to the user to create an assigned desktop. A desktop session is then launched to the machine. Subsequent launches are routed directly to the now assigned machine.

Once a machine has been assigned in this way, the original assignment rule plays no further part in access to the new desktop.

Multiple desktop rules in the assignment policy can apply to the same desktop group. Where a user is granted entitlements by more than one rule for the same group, they can have as many machine assignments from the group as the total of their entitlements.

## Related topics

[Get-BrokerAssignmentPolicyRule](#)

[Set-BrokerAssignmentPolicyRule](#)

[Rename-BrokerAssignmentPolicyRule](#)

[Remove-BrokerAssignmentPolicyRule](#)

## Parameters

**-Name**<String>

Specifies the administrative name of the new desktop rule. Each rule in the site's assignment policy must have a unique name (irrespective of whether they are desktop or application rules).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopGroupUid<Int32>**

Specifies the unique ID of the desktop group to which the new desktop rule applies.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ColorDepth<ColorDepth>**

Specifies the color depth of any desktop sessions to machines assigned by the new rule.

Valid values are \$null, FourBit, EightBit, SixteenBit, and TwentyFourBit.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |                                                     |
|------------------------|-----------------------------------------------------|
| Required?              | false                                               |
| Default Value          | null (dynamically inherited from the desktop group) |
| Accept Pipeline Input? | true (ByPropertyName)                               |

**-Description<String>**

Specifies an optional description of the new desktop rule. The text may be visible to the end user, for example, as a tooltip associated with the desktop entitlement.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |                                                     |
|------------------------|-----------------------------------------------------|
| Required?              | false                                               |
| Default Value          | null (dynamically inherited from the desktop group) |
| Accept Pipeline Input? | true (ByPropertyName)                               |

**-Enabled<Boolean>**

Specifies whether the new desktop rule is initially enabled. A disabled rule is ignored when evaluating the site's assignment policy.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedUserFilterEnabled<Boolean>**

Specifies whether the excluded users filter is initially enabled. If the filter is disabled then any user entries in the filter are ignored when

assignment policy rules are evaluated.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedUsers<User[]>**

Specifies the excluded users filter of the new desktop rule, that is, the users and groups who are explicitly denied an entitlement to a machine assignment from the rule.

This can be used to exclude users or groups who would otherwise gain access by groups specified in the included users filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IconUid<Int32>**

Specifies the unique ID of the icon used to display the machine assignment entitlement to the user, and of the assigned desktop itself following the assignment.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |                                                     |
|------------------------|-----------------------------------------------------|
| Required?              | false                                               |
| Default Value          | null (dynamically inherited from the desktop group) |
| Accept Pipeline Input? | true (ByPropertyName)                               |

**-IncludedUserFilterEnabled<Boolean>**

Specifies whether the included users filter is initially enabled. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to a machine assignment by the new desktop rule.

Users who would be implicitly granted access when the filter is disabled can still be explicitly denied access using the excluded users filter.

For rules that relate to RemotePC desktop groups however, if the included user filter is disabled, the rule is effectively disabled.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedUsers<User[]>**

Specifies the included users filter of the new desktop rule, that is, the users and groups who are granted an entitlement to a machine assignment by the rule.

If a user appears explicitly in the excluded users filter of the rule or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MaxDesktops**<Int32>

The number of machines from the rule's desktop group to which a user is entitled. Where an entitlement is granted to a user group rather than an individual, the number of machines applies to each member of the user group independently.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 1                     |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PublishedName**<String>

The name of the new machine assignment entitlement as seen by the user, and of the assigned desktop following its usage.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |                                                     |
|------------------------|-----------------------------------------------------|
| Required?              | false                                               |
| Default Value          | null (dynamically inherited from the desktop group) |
| Accept Pipeline Input? | true (ByPropertyName)                               |

**-SecureIcaRequired**<Boolean>

Specifies whether the new desktop rule requires the SecureICA protocol to be used for desktop sessions to machines assigned using the entitlement.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |                                                     |
|------------------------|-----------------------------------------------------|
| Required?              | false                                               |
| Default Value          | null (dynamically inherited from the desktop group) |
| Accept Pipeline Input? | true (ByPropertyName)                               |

**-UUID**<Guid>



An optional GUID for this rule.

|                        |                                              |
|------------------------|----------------------------------------------|
| Required?              | false                                        |
| Default Value          | A new GUID is generated if none is supplied. |
| Accept Pipeline Input? | true (ByPropertyName)                        |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.AssignmentPolicyRule

New-BrokerAssignmentPolicyRule returns the newly created desktop rule in the assignment policy rule.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Sales Support'  
C:\PS> New-BrokerAssignmentPolicyRule 'UK Office' -DesktopGroupUid $dg.Uid -IncludedUsers sales\uk-staff -PublishedName 'Sales Desktop'  
Creates a desktop rule in the assignment policy that grants all members of the SALES\uk-staff group an entitlement to a single machine from the Sales Support desktop group. The entitlement name seen by users is Sales Desktop.
```

# New-BrokerCatalog

Jan 04, 2017

Adds a new catalog to the site.

## Syntax

```
New-BrokerCatalog [-Name] <String> [-AllocationType] <AllocationType> [-CatalogKind] <CatalogKind> [-PvsForVM <String[]>] [-Description <String>] [-IsRemotePC <Boolean>] [-MachinesArePhysical <Boolean>] [-MinimumFunctionalLevel <FunctionalLevel>] [-PvsAddress <String>] [-PvsDomain <String>] [-RemotePCHypervisorConnectionUid <Int32>] [-Scope <String[]>] [-UUID <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-BrokerCatalog [-Name] <String> [-AllocationType] <AllocationType> [-ProvisioningType] <ProvisioningType> [-SessionSupport] <SessionSupport> [-PersistUserChanges] <PersistUserChanges> [-ProvisioningSchemeId <Guid>] [-Description <String>] [-IsRemotePC <Boolean>] [-MachinesArePhysical <Boolean>] [-MinimumFunctionalLevel <FunctionalLevel>] [-PvsAddress <String>] [-PvsDomain <String>] [-RemotePCHypervisorConnectionUid <Int32>] [-Scope <String[]>] [-UUID <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

New-BrokerCatalog adds a catalog through which machines can be provided to the site.

In order for a machine to register in a site, the machine must belong to a catalog with which it is compatible. The compatibility of a machine with a catalog is determined by two of the parameters of New-BrokerCatalog:

o MinimalFunctionalLevel: The minimal functional level supported in the catalog. The functional level of the machine is determined by the capabilities of the Citrix VDA software on it.

o SessionSupport: The session support (single/multi) of the catalog. The session support of the machine is determined by the variant of the Citrix VDA software installed (workstation/terminal services, respectively).

## Related topics

[Get-BrokerCatalog](#)

[Rename-BrokerCatalog](#)

[Remove-BrokerCatalog](#)

[Set-BrokerCatalog](#)

## Parameters

**-Name**<String>

Specifies a name for the catalog. Each catalog within a site must have a unique name.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-AllocationType**<AllocationType>

Specifies how machines in the catalog are assigned to users. Values can be:

o Static - Machines in a catalog of this type are permanently assigned to a user.

o Permanent - equivalent to 'Static'.

o Random - Machines in a catalog of this type are picked at random and temporarily assigned to a user.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-CatalogKind**<CatalogKind>

Deprecated: The type of machines the catalog will contain. Values can be: ThinCloned, SingleImage, PowerManaged, Unmanaged, Pvs, Pvd or PvsPvd.

Thin-Cloned, Single-Image and Personal vDisk Catalogs

Thin-cloned and single-image catalog kinds are for machines created and managed with Provisioning Services for VMs. All machines in this type of catalog are managed, and so must be associated with a hypervisor connection.

A thin-cloned catalog is used for original golden VM images that are cloned when they are assigned to a VM, and users' changes to the VM image are retained after the VM is restarted.

A single-image catalog is used when multiple machines provisioned with Provisioning Services for VMs all share a single golden VM image when they run and, when restarted, they revert to the original VM image state.

A personal vDisk catalog is similar to a single-image catalog, but it also uses personal vDisk technology.

**PowerManaged**

-----

This catalog kind is for managed machines that are manually provisioned by administrators. All machines in this type of catalog are managed, and so must be associated with a hypervisor connection.

**Unmanaged**

-----

This catalog kind is for unmanaged machines, so there is no associated hypervisor connection.

**PVS**

---

This catalog kind is for managed machines that are provisioned using Provisioning Services. All machines in this type of catalog are managed, and so must be associated with a hypervisor connection. Only shared desktops are suitable for this catalog kind.

A Provisioning Services-personal vDisk (PvsPvd) catalog is similar to a Provisioning Services catalog, but it also uses personal vDisk technology.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ProvisioningType**<ProvisioningType>

Specifies the ProvisioningType for the catalog. Values can be:

- o Manual - No provisioning.
- o PVS - Machine provisioned by PVS (machine may be physical, blade, VM,...).
- o MCS - Machine provisioned by MCS (machine must be VM).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-SessionSupport**<SessionSupport>

Specifies whether machines in the catalog are single or multi-session capable. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PersistUserChanges**<PersistUserChanges>

Specifies how user changes are persisted on machines in the catalog. Possible values are:

- o OnLocal: User changes are stored on the machine's local storage.
- o Discard: User changes are discarded.
- o OnPvd: User changes are stored on the user's personal vDisk.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PvsForVM**<String[]>

Deprecated:

Identifies the provisioning scheme used by this catalog. To be specified in the format: ProvisioningSchemeGuid:ServiceGroupGuid. Applicable only to thin-cloned, single-image or personal vDisk catalogs.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Description**<String>

A description for the catalog.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IsRemotePC**<Boolean>

Specifies whether this is to be a Remote PC catalog.

IsRemotePC can only be enabled when:

o SessionSupport is SingleSession

o MachinesArePhysical is true.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MachinesArePhysical**<Boolean>

Specifies whether machines in the catalog can be power-managed by the Citrix Broker Service. Where the Citrix Broker Service cannot control the power state of themachine specify True, otherwise False. Can only be specified together with a provisioning type of Pvs or Manual, or if used with the legacy CatalogKind parameter only with Pvs or PvsPvd catalog kinds.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MinimumFunctionalLevel**<FunctionalLevel>

The minimum FunctionalLevel required for machines to register in the site.

Valid values are L5, L7, L7\_6

|                        |                                                                                                                                               |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Required?              | false                                                                                                                                         |
| Default Value          | The FunctionalLevel of the current release (L7_6); by default no machines with less than the most current FunctionalLevel will be functional. |
| Accept Pipeline Input? | true (ByPropertyName)                                                                                                                         |

**-PvsAddress**<String>

Specifies the URL of the Provisioning Services server. Only applicable to Provisioning Services or Provisioning Services-personal vDisk catalogs.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PvsDomain<String>**

Specifies the Active Directory domain of the Provisioning Services server. Only applicable to Provisioning Services or Provisioning Services-personal vDisk catalogs.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-RemotePCHypervisorConnectionUid<Int32>**

Specifies the hypervisor connection to use for powering on remote PCs in this catalog (only allowed when IsRemotePC is true).

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Scope<String[]>**

Specifies the name of the delegated administration scope to which the catalog belongs.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-UUID<Guid>**

An optional GUID for this catalog.

|                        |                                              |
|------------------------|----------------------------------------------|
| Required?              | false                                        |
| Default Value          | A new GUID is generated if none is supplied. |
| Accept Pipeline Input? | true (ByPropertyName)                        |

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**-ProvisioningSchemeId<Guid>**

Specifies the identity of the MCS provisioning scheme the new catalog is associated with (can only be specified for new catalogs with a ProvisioningType of MCS).

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

| Default Value          | Null                  |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |

#### Input Type

None You cannot pipe input into this cmdlet.

#### Return Values

Citrix.Broker.Admin.SDK.Catalog

New-BrokerCatalog returns the created catalog.

#### Examples

##### ----- EXAMPLE 1 -----

C:\PS> New-BrokerCatalog -AllocationType Static -CatalogKind Unmanaged -Description "Catalog1 Description" -Name "Catalog1 Name"  
 This command creates a catalog that can contain unmanaged physical or virtual machines that are permanently assigned to the user.

##### ----- EXAMPLE 2 -----

C:\PS> New-BrokerCatalog -AllocationType Random -CatalogKind PowerManaged -Description "catalog 2 Description" -Name "Catalog2 Name"  
 This command creates a catalog that can contain power-managed machines that are randomly assigned to the user.

##### ----- EXAMPLE 3 -----

C:\PS> New-BrokerCatalog -AllocationType Random -CatalogKind PVS -Description "PVS Catalog Desc" -Name "PVS Catalog Name" -PvsAddress "pvsServer@pvsDomain.com" -PvsDomain "pvsDor"  
 This command creates a catalog that can contain managed machines that are provisioned using Provisioning Services.

# New-BrokerConfigurationSlot

Jan 04, 2017

Creates a new configuration slot.

## Syntax

```
New-BrokerConfigurationSlot [-Name] <String> -SettingsGroup <String> [-Description <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Creates a new configuration slot. The SettingsGroup of the slot determines the particular collection of related settings that may be specified in a machine configuration associated with this slot.

For example, the configuration slot may be restricted to configuring Citrix User Profile Manager settings by specifying the SettingsGroup parameter as "G=UPM".

## Related topics

[Get-BrokerConfigurationSlot](#)

[Remove-BrokerConfigurationSlot](#)

[New-BrokerMachineConfiguration](#)

## Parameters

**-Name**<String>

Name of the new configuration slot. This must be alphanumeric and not contain white space.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-SettingsGroup**<String>

The settings group determines the particular collection of related settings that may be controlled by this slot. This must match the format of a Citrix Group Policy configuration group (e.g. "G=UPM"). Only settings that have this exact group may be specified in a machine configuration associated with this configuration slot.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-Description**<String>

Description of configuration slot.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.ConfigurationSlot

New-BrokerConfigurationSlot returns an object representing the newly created configuration slot



## Examples

### ----- EXAMPLE 1 -----

```
New-BrokerConfigurationSlot -Name "UPM" -SettingsGroup "G=UPM"
```

Create a new slot named "UPM" to configure settings specific to "User Profile Management"

# New-BrokerConfiguredFTA

Jan 04, 2017

Creates a file type association with a published application.

## Syntax

```
New-BrokerConfiguredFTA -ImportedFTA <ImportedFTA> -ApplicationUid <Int32> [-UUID <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [-CommonParameters <String>]
```

```
New-BrokerConfiguredFTA -ExtensionName <String> -HandlerName <String> -ApplicationUid <Int32> [-ContentType <String>] [-HandlerDescription <String>] [-HandlerOpenArguments <String>] [-UUID <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [-CommonParameters <String>]
```

## Detailed Description

Creates an association between a file type and a published application for the purposes of the content redirection.

File type association associates a file extension (such as ".txt") with an application (such as Notepad). In a Citrix environment file type associations on a user device can be configured so that when an user clicks on a document it launches the appropriate published application. This is known as "content redirection".

Configured file type associations are different from imported file type associations. Configured file type associations are those that are actually associated with published applications for the purposes of content redirection. Imported file type associations are lists of known file type associations for a given desktop group. See Update-BrokerImportedFTA for more information about imported file type associations.

This cmdlet has two parameter sets, which correspond to the cmdlet's two use cases.

The first use case leverages imported file type associations to configure file types for published applications. Information about the file type association is read from the imported object. See the Update-BrokerImportedFTA cmdlet for more information about importing file type associations from a worker machine.

The second use case is more complex and allows you to create your own file type association without having to import it first. This also lets you create custom file type associations that may not already exist on the worker machines. This use case is more error-prone, however, because the individual attributes of the file type association must be correctly specified by you.

## Related topics

[Get-BrokerImportedFTA](#)

[Get-BrokerConfiguredFTA](#)

[Remove-BrokerConfiguredFTA](#)

## Parameters

**-ApplicationUid**<Int32>

Specifies the application with which the file type should be associated.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | (required)            |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ImportedFTA**<ImportedFTA>

Specifies the ImportedFTA object to use for creating the ConfiguredFTA object. All values needed to create a ConfiguredFTA object are read from the ImportedFTA object.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | (required)            |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExtensionName**<String>

Specifies the extension name for the file type association. For example, ".txt" or ".doc".

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | (required)            |
| Accept Pipeline Input? | true (ByPropertyName) |

**-HandlerName**<String>

Specifies the name of the handler for the file type association (as seen in the Registry). For example, "TXTFILE" or "Word.Document.8".

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          | (required)            |
| Accept Pipeline Input? | true (ByPropertyName) |

**-UUID<Guid>**

An optional GUID for this ConfiguredFTA.

|                        |                                              |
|------------------------|----------------------------------------------|
| Required?              | false                                        |
| Default Value          | A new GUID is generated if none is supplied. |
| Accept Pipeline Input? | false                                        |

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**-ContentType<String>**

Specifies the content type of the file type (as listed in the Registry). For example, content type would be "text/plain" or "application/msword".

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-HandlerDescription<String>**

Specifies the description of the handler for the file type association.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-HandlerOpenArguments<String>**

Specifies the arguments for the open command that the handler should use. For example, "%1".

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

## Input Type

Variable, based on property name. This cmdlet does accept input from the pipeline but only by property name.

## Return Values

Citrix.Broker.Admin.SDK.ConfiguredFTA

This cmdlet returns a single ConfiguredFTA object.

## Examples

### EXAMPLE 1

```
C:\PS> $app = Get-BrokerApplication "Notepad"
```

```
C:\PS> $fta = Get-BrokerImportedFTA -ExtensionName ".txt"
```

```
C:\PS> New-BrokerConfiguredFTA -ImportedFTA $fta -ApplicationUid $app.Uid
```

Gets the Uid for the application, gets the ImportedFTA object for the file extension, and finally associates ".txt" with the published "Notepad" application.

Note that the Get-BrokerImportedFTA cmdlet may return more than one ImportedFTA objects for a specific extension name. See the help for that cmdlet for more details.

### EXAMPLE 2

```
C:\PS> $app = Get-BrokerApplication "Notepad"
```

```
C:\PS> New-BrokerConfiguredFTA -ApplicationUid $app.Uid -ExtensionName ".txt" -HandlerName "txtfile" -ContentType "text/plain" -HandlerDescription "Text Document" -HandlerOpenArguments "%1"
```

This example is identical to the first, but shows the the second use case of the cmdlet, specifying each attribute manually.

# New-BrokerDelayedHostingPowerAction

Jan 04, 2017

Causes a power action to be queued after a delay.

## Syntax

```
New-BrokerDelayedHostingPowerAction [-MachineName] <String> -Action <PowerManagementAction> -Delay <TimeSpan> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Causes a power action to be queued after the specified period of time.

Only ShutDown or Suspend actions can be requested to be delayed in this manner.

For a detailed description of the queuing mechanism, see 'help about\_Broker\_PowerManagement'.

## Related topics

[Get-BrokerDelayedHostingPowerAction](#)

[New-BrokerDelayedHostingPowerAction](#)

## Parameters

**-MachineName**<String>

Specifies the machine that the action is to be performed on.

The machine can be identified by DNS name, short name, SID, or name of the form domain\machine.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Action**<PowerManagementAction>

Specifies the power state change action that is to be performed on the specified machine after the specified delay.

Valid values are Shutdown and Suspend.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

### **-Delay**<TimeSpan>

Specifies a timespan delay before the action is queued.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

None You cannot pipe input into this cmdlet.

### Return Values

Citrix.Broker.Admin.SDK.DelayedHostingPowerAction

New-BrokerDelayedHostingPowerAction returns the created delayed power action.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> New-BrokerDelayedHostingPowerAction -Action Shutdown -MachineName 'XD_VDA1' -Delay '00:02:00'
```

Causes the machine called XD\_VDA1 to be shut down after a delay of two minutes.

# New-BrokerDesktopGroup

Jan 04, 2017

Create a new desktop group for managing the brokering of groups of desktops.

## Syntax

```
New-BrokerDesktopGroup [-Name] <String> -DesktopKind <DesktopKind> [-AutomaticPowerOnForAssigned <Boolean>] [-AutomaticPowerOnForAssignedDuringPeak <Boolean>] [-ColorDepth <ColorDepth>] [-DeliveryType <DeliveryType>] [-Description <String>] [-Enabled <Boolean>] [-IconUid <Int32>] [-InMaintenanceMode <Boolean>] [-IsRemotePC <Boolean>] [-MinimumFunctionalLevel <FunctionalLevel>] [-OffPeakBufferSizePercent <Int32>] [-OffPeakDisconnectAction <SessionChangeHostingAction>] [-OffPeakDisconnectTimeout <Int32>] [-OffPeakExtendedDisconnectAction <SessionChangeHostingAction>] [-OffPeakExtendedDisconnectTimeout <Int32>] [-OffPeakLogOffAction <SessionChangeHostingAction>] [-OffPeakLogOffTimeout <Int32>] [-PeakBufferSizePercent <Int32>] [-PeakDisconnectAction <SessionChangeHostingAction>] [-PeakDisconnectTimeout <Int32>] [-PeakExtendedDisconnectAction <SessionChangeHostingAction>] [-PeakExtendedDisconnectTimeout <Int32>] [-PeakLogOffAction <SessionChangeHostingAction>] [-PeakLogOffTimeout <Int32>] [-ProtocolPriority <String[]>] [-PublishedName <String>] [-Scope <String[]>] [-SecureIcaRequired <Boolean>] [-SessionSupport <SessionSupport>] [-SettlementPeriodBeforeAutoShutdown <TimeSpan>] [-ShutdownDesktopsAfterUse <Boolean>] [-TimeZone <String>] [-TurnOnAddedMachine <Boolean>] [-UUID <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The New-BrokerDesktopGroup cmdlet creates a new broker desktop group that can then be used to manage the brokering settings of all desktops within that desktop group. Once the desktop group has been created, you can create desktops in it by adding the appropriate broker machines to it using the Add-BrokerMachine or Add-BrokerMachinesToDesktopGroup cmdlets.

Desktop groups hold settings that apply to all desktops they contain.

For any automatic power management settings of a desktop group to take effect, the group's TimeZone property must be specified. Automatic power management operations include pool management (power time schemes), reboot schedules, session disconnect and logoff actions, and powering on assigned machines etc.

## Related topics

[Get-BrokerDesktopGroup](#)

[Set-BrokerDesktopGroup](#)

[Rename-BrokerDesktopGroup](#)

[Remove-BrokerDesktopGroup](#)

[Add-BrokerMachine](#)

[Add-BrokerMachinesToDesktopGroup](#)

[Get-BrokerSite](#)

## Parameters



**-Name**<String>

The name of the new broker desktop group.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopKind**<DesktopKind>

The kind of desktops this group will hold. Valid values are Private and Shared.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-AutomaticPowerOnForAssigned**<Boolean>

Specifies whether assigned desktops in the desktop group should be automatically started at the start of peak time periods. Only relevant for groups whose DesktopKind is Private.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-AutomaticPowerOnForAssignedDuringPeak**<Boolean>

Specifies whether assigned desktops in the desktop group should be automatically started throughout peak time periods. Only relevant for groups whose DesktopKind is Private and which have AutomaticPowerOnForAssigned set to true.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ColorDepth**<ColorDepth>

Specifies the color depth that the ICA session should use for desktops in this group. Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | TwentyFourBit         |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DeliveryType**<DeliveryType>

Specifies whether desktops, applications, or both, can be delivered from machines contained within the new desktop group. Desktop groups with a DesktopKind of Private cannot be used to deliver both desktops and applications. Defaults to DesktopsOnly if not specified.

Valid values are DesktopsOnly, AppsOnly, and DesktopsAndApps.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | DesktopsOnly          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Description**<String>

A description for this desktop group useful for administrators of the site.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Enabled**<Boolean>

Whether the desktop group should be in the enabled state; disabled desktop groups do not appear to users.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | true  |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-IconUid**<Int32>

The UID of the broker icon to be displayed to users for their desktop(s) in this desktop group.

|                        |                                                                                                      |
|------------------------|------------------------------------------------------------------------------------------------------|
| Required?              | false                                                                                                |
| Default Value          | The Uid of the default desktop icon in this site - use the Get-BrokerSite cmdlet to find this value. |
| Accept Pipeline Input? | true (ByPropertyName)                                                                                |

**-InMaintenanceMode**<Boolean>

Whether the desktop should be created in maintenance mode; a desktop group in maintenance mode will not allow users to connect or reconnect to their desktops.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IsRemotePC**<Boolean>

Specifies whether this is to be a Remote PC desktop group.

IsRemotePC can only be enabled when:

- o SessionSupport is SingleSession
- o DeliveryType is DesktopsOnly
- o DesktopKind is Private

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MinimumFunctionalLevel**<FunctionalLevel>

The minimum FunctionalLevel required for machines to work successfully in the desktop group.

Valid values are L5, L7, L7\_6

|                        |                                                                                                                                               |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Required?              | false                                                                                                                                         |
| Default Value          | The FunctionalLevel of the current release (L7_6); by default no machines with less than the most current FunctionalLevel will be functional. |
| Accept Pipeline Input? | true (ByPropertyName)                                                                                                                         |

**-OffPeakBufferSizePercent**<Int32>

The percentage of machines in the desktop group that should be kept available in an idle state outside peak hours.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 0                     |
| Accept Pipeline Input? | true (ByPropertyName) |

**-OffPeakDisconnectAction**<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session disconnecting outside peak hours. Possible values are Nothing, Suspend, or Shutdown

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | Nothing               |
| Accept Pipeline Input? | true (ByPropertyName) |

**-OffPeakDisconnectTimeout**<Int32>

The number of minutes before the configured action should be performed after a user session disconnects outside peak hours.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 0                     |
| Accept Pipeline Input? | true (ByPropertyName) |

---

**-OffPeakExtendedDisconnectAction**<SessionChangeHostingAction>

The action to be performed after a second configurable period of a user session disconnecting outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | Nothing               |
| Accept Pipeline Input? | true (ByPropertyName) |

**-OffPeakExtendedDisconnectTimeout**<Int32>

The number of minutes before the second configured action should be performed after a user session disconnects outside peak hours.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 0                     |
| Accept Pipeline Input? | true (ByPropertyName) |

**-OffPeakLogOffAction**<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session ending outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | Nothing               |
| Accept Pipeline Input? | true (ByPropertyName) |

**-OffPeakLogOffTimeout**<Int32>

The number of minutes before the configured action should be performed after a user session ends outside peak hours.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? |       |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-PeakBufferSizePercent**<Int32>

The percentage of machines in the desktop group that should be kept available in an idle state in peak hours.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 0                     |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PeakDisconnectAction**<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session disconnecting in peak hours. Possible values are Nothing, Suspend, or Shutdown.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | Nothing               |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PeakDisconnectTimeout**<Int32>

The number of minutes before the configured action should be performed after a user session disconnects in peak hours.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 0                     |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PeakExtendedDisconnectAction**<SessionChangeHostingAction>

The action to be performed after a second configurable period of a user session disconnecting in peak hours. Possible values are Nothing, Suspend, or Shutdown.

|               |         |
|---------------|---------|
| Required?     | false   |
| Default Value | Nothing |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-PeakExtendedDisconnectTimeout<Int32>**

The number of minutes before the second configured action should be performed after a user session disconnects in peak hours.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 0                     |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PeakLogOffAction<SessionChangeHostingAction>**

The action to be performed after a configurable period of a user session ending in peak hours. Possible values are Nothing, Suspend, or Shutdown.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | Nothing               |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PeakLogOffTimeout<Int32>**

The number of minutes before the configured action should be performed after a user session ends in peak hours.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 0                     |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ProtocolPriority<String[]>**

A list of protocol names in the order in which they should be attempted for use during connection.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | null  |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-PublishedName**<String>

The name that will be displayed to users for their desktop(s) in this desktop group.

|                        |                                                                    |
|------------------------|--------------------------------------------------------------------|
| Required?              | false                                                              |
| Default Value          | The same value as that supplied for the name of the desktop group. |
| Accept Pipeline Input? | true (ByPropertyName)                                              |

**-Scope**<String[]>

Specifies the name of the delegated administration scope to which the desktop group should belong.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-SecureIcaRequired**<Boolean>

Whether HDX connections to desktops in the new desktop group require the use of a secure protocol.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-SessionSupport**<SessionSupport>

Specifies whether machines in the desktop group are single or multi-session capable. Values can be:

- o SingleSession - Single-session only machine.
- o MultiSession - Multi-session capable machine.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |



|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-SettlementPeriodBeforeAutoShutdown**<TimeSpan>

Time after a session ends during which automatic shutdown requests (for example, shutdown after use, idle pool management) are deferred. Any outstanding shutdown request takes effect after the settlement period expires. This is typically used to configure time to allow logoff scripts to complete.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 0                     |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ShutdownDesktopsAfterUse**<Boolean>

Whether desktops in this desktop group should be automatically shut down when each user session completes (only relevant to power-managed desktops).

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-TimeZone**<String>

The time zone in which this desktop group's machines reside.

The time zone must be specified for any of the group's automatic power management settings to take effect. Automatic power management operations include pool management (power time schemes), reboot schedules, session disconnect and logoff actions, and powering on assigned machines etc.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-TurnOnAddedMachine**<Boolean>

This flag specifies whether the Broker Service should attempt to power on machines when they are added to the desktop

group.

|                        |                                                                            |
|------------------------|----------------------------------------------------------------------------|
| Required?              | false                                                                      |
| Default Value          | \$false for single session machines and \$true for multi-session machines. |
| Accept Pipeline Input? | true (ByPropertyName)                                                      |

**-UUID**<Guid>

An optional GUID for this desktop group.

|                        |                                              |
|------------------------|----------------------------------------------|
| Required?              | false                                        |
| Default Value          | A new GUID is generated if none is supplied. |
| Accept Pipeline Input? | true (ByPropertyName)                        |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

Citrix.Broker.Admin.SDK.DesktopGroup

The newly created desktop group.

## Notes

Once a new desktop group is created, you can create desktops in it by adding the appropriate broker machines to it using the Add-BrokerMachine or Add-BrokerMachinesToDesktopGroup cmdlets.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> New-BrokerDesktopGroup "Assigned Desktops" -PublishedName "MyDesktop" -DesktopKind Private
Create a desktop group to manage the brokering of private desktops, which will appear to users with the name "MyDesktop".
```

# New-BrokerEntitlementPolicyRule

Jan 04, 2017

Creates a new desktop rule in the site's entitlement policy.

## Syntax

```
New-BrokerEntitlementPolicyRule [-Name] <String> -DesktopGroupUid <Int32> [-ColorDepth <ColorDepth>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IconUid <Int32>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-PublishedName <String>] [-SecureIcaRequired <Boolean>] [-SessionReconnection <SessionReconnection>] [-UUID <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The New-BrokerEntitlementPolicyRule cmdlet adds a new desktop rule to the site's entitlement policy.

A desktop rule in the entitlement policy defines the users who are allowed per-session access to a machine from the rule's associated desktop group to run a full desktop session.

The following constraints apply when creating a desktop entitlement rule for a desktop group:

- o The group's desktop kind must be Shared
- o The group's delivery type must be DesktopsOnly or DesktopsAndApps

When a user selects a desktop entitlement published from a shared group, a machine is selected from the group on which to run the desktop session. No permanent association exists between the user and the selected machine; once the session ends the association also ends.

Multiple desktop rules in the entitlement policy can apply to the same desktop group. Where a user is granted an entitlement by more than one rule for the same group, they can use as many desktop sessions at the same time as they have entitlements.

## Related topics

[Get-BrokerEntitlementPolicyRule](#)

[Set-BrokerEntitlementPolicyRule](#)

[Rename-BrokerEntitlementPolicyRule](#)

[Remove-BrokerEntitlementPolicyRule](#)

## Parameters

**-Name**<String>

Specifies the administrative name of the new desktop rule. Each rule in the site's entitlement policy must have a unique name (irrespective of whether they are desktop or application rules).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopGroupUid**<Int32>

Specifies the unique ID of the desktop group to which the new desktop rule applies.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ColorDepth**<ColorDepth>

Specifies the color depth of any desktop sessions launched by a user from this entitlement.

Valid values are \$null, FourBit, EightBit, SixteenBit, and TwentyFourBit.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |                                                     |
|------------------------|-----------------------------------------------------|
| Required?              | false                                               |
| Default Value          | null (dynamically inherited from the desktop group) |
| Accept Pipeline Input? | true (ByPropertyName)                               |

**-Description**<String>

Specifies an optional description of the new desktop rule. The text may be visible to the end user, for example, as a tooltip associated with the desktop entitlement.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |                                                     |
|------------------------|-----------------------------------------------------|
| Required?              | false                                               |
| Default Value          | null (dynamically inherited from the desktop group) |
| Accept Pipeline Input? | true (ByPropertyName)                               |

**-Enabled**<Boolean>

Specifies whether the new desktop rule is initially enabled. A disabled rule is ignored when evaluating the site's entitlement policy.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ExcludedUserFilterEnabled**<Boolean>

Specifies whether the excluded users filter is initially enabled. If the filter is disabled then any user entries in the filter are ignored when entitlement policy rules are evaluated.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | false |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-ExcludedUsers<User[]>**

Specifies the excluded users filter of the desktop rule, that is, the users and groups who are explicitly denied an entitlement to a desktop session from the new rule.

This can be used to exclude users or groups who would otherwise gain access by groups specified in the included users filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IconUid<Int32>**

Specifies the unique ID of the icon used to display the desktop session entitlement to the user.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |                                                     |
|------------------------|-----------------------------------------------------|
| Required?              | false                                               |
| Default Value          | null (dynamically inherited from the desktop group) |
| Accept Pipeline Input? | true (ByPropertyName)                               |

**-IncludedUserFilterEnabled<Boolean>**

Specifies whether the included users filter is initially enabled. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to a desktop session by the new rule.

Users who would be implicitly granted access when the filter is disabled can still be explicitly denied access using the excluded users filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-IncludedUsers<User[]>**

Specifies the included users filter of the rule, that is, the users and groups who are granted an entitlement to a desktop session by the new rule.

If a user appears explicitly in the excluded users filter of the rule or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | (empty list)          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PublishedName**<String>

The name of the new desktop session entitlement as seen by the user.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |                                                     |
|------------------------|-----------------------------------------------------|
| Required?              | false                                               |
| Default Value          | null (dynamically inherited from the desktop group) |
| Accept Pipeline Input? | true (ByPropertyName)                               |

**-SecureIcaRequired**<Boolean>

Specifies whether the new desktop rule requires the SecureICA protocol for desktop sessions launched using the entitlement.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |                                                     |
|------------------------|-----------------------------------------------------|
| Required?              | false                                               |
| Default Value          | null (dynamically inherited from the desktop group) |
| Accept Pipeline Input? | true (ByPropertyName)                               |

**-SessionReconnection**<SessionReconnection>

Defines reconnection (roaming) behavior for sessions launched using this rule. Session reconnection control is an experimental and unsupported feature.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | Always                |
| Accept Pipeline Input? | true (ByPropertyName) |

**-UUID**<Guid>

An optional GUID for this rule.

|                        |                                              |
|------------------------|----------------------------------------------|
| Required?              | false                                        |
| Default Value          | A new GUID is generated if none is supplied. |
| Accept Pipeline Input? | true (ByPropertyName)                        |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation

and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.EntitlementPolicyRule

New-BrokerEntitlementPolicyRule returns the newly created desktop rule in the entitlement policy.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Customer Support'
C:\PS> New-BrokerEntitlementPolicyRule 'UK Office' -DesktopGroupUid $dg.Uid -IncludedUsers support\uk-staff -PublishedName 'Support Desktop'
Creates an desktop rule in the entitlement policy that entitles all members of the SUPPORT\uk-staff group to a desktop session from the Customer Support desktop group. The desktop entitlement name seen by users is Support Desktop.
```



# New-BrokerHostingPowerAction

Jan 04, 2017

Creates a new action in the power action queue.

## Syntax

```
New-BrokerHostingPowerAction [-MachineName] <String> -Action <PowerManagementAction> [-ActualPriority <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The New-BrokerHostingPowerAction cmdlet adds a new power action record into the queue of power actions to be performed. The power actions in the queue are processed on a priority basis and sent to the relevant hypervisor to change the power state of a virtual machine.

A power action record defines the action to be performed, the machine on which the action is to be performed, and an initial priority value for the action. Multiple actions may be created that relate to the same machine.

For a detailed description of the queuing mechanism, see 'help about\_Broker\_PowerManagement'.

## Related topics

[Get-BrokerHostingPowerAction](#)

[Set-BrokerHostingPowerAction](#)

[Remove-BrokerHostingPowerAction](#)

## Parameters

**-MachineName**<String>

Specifies the machine that the action is to be performed on.

The machine can be identified by DNS name or short name or SID or 'machine\domain' form name.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Action**<PowerManagementAction>

Specifies the power state change action that is to be performed on the specified machine.

Valid values are: TurnOn, TurnOff, ShutDown, Reset, Restart, Suspend and Resume.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ActualPriority**<Int32>

Specifies an initial priority value for the action in the queue.

This priority is the current action priority; the 'base' priority for actions created via this cmdlet is always 30. Numerically lower priority values indicate more important actions that are processed in preference to actions with numerically higher priority settings.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 30                    |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

Citrix.Broker.Admin.SDK.HostingPowerAction

New-BrokerHostingPowerAction returns the newly created power action record.

## Examples

### ----- **EXAMPLE 1** -----

```
C:\PS> New-BrokerHostingPowerAction -Action Shutdown -MachineName 'XD_VDA1'
```

Causes the machine called 'XD\_VDA1' to be shut down.

# New-BrokerHypervisorConnection

Jan 04, 2017

Creates a new hypervisor connection.

## Syntax

```
New-BrokerHypervisorConnection [-HypHypervisorConnectionUid] <Guid> [-PreferredController <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [-CommonParameters<>]
```

## Detailed Description

The New-BrokerHypervisorConnection cmdlet creates a new hypervisor connection.

## Related topics

[Get-BrokerHypervisorConnection](#)

[Remove-BrokerHypervisorConnection](#)

[Set-BrokerHypervisorConnection](#)

## Parameters

### **-HypHypervisorConnectionUid**<Guid>

The Guid that identifies the hypervisor connection, as defined in DUM.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

### **-PreferredController**<String>

The preferred controller machine for the hypervisor connection. Can be specified as (first match is used):

- o Full SAM name.
- o Full DNS name.
- o SID value.
- o NetBIOS name (SAM without domain).
- o Partial DNS name (DNS name without some or all domain information).

Where not specified, the system selects preferred controller machine based on loading.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None

Return Values

Citrix.Broker.Admin.SDK.HypervisorConnection

New-BrokerHypervisorConnection returns an opaque object containing information about the hypervisor connection.

Examples

----- **EXAMPLE 1** -----

C:\PS> New-BrokerHypervisorConnection -PreferredController "domainName\controllerName" -HypHypervisorConnectionUid "d16f4e56-b85e-4ba6-b745-0e978ae4f192"  
This command creates a new hypervisor connection with a preferred controller.

----- **EXAMPLE 2** -----

C:\PS> New-BrokerHypervisorConnection -HypHypervisorConnectionUid "d16f4e56-b85e-4ba6-b745-0e978ae4f192"  
This command creates a new hypervisor connection, and leaves it to the system to select a preferred controller.

# New-BrokerIcon

Jan 04, 2017

Creates a new icon.

## Syntax

```
New-BrokerIcon [-EncodedIconData] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Accepts Base64 encoded .ICO format icon data, stores it in the database and returns an Icon object containing the Uid assigned to it.

New-BrokerIcon can be used with the Get-CtxIcon cmdlet from Citrix.Common.Commands, to obtain the Base64 icon. See Examples for a demonstration.

## Related topics

[Get-CtxIcon](#)

[Get-BrokerIcon](#)

[Remove-BrokerIcon](#)

## Parameters

**-EncodedIconData**<String>

Specifies the Base64 encoded .ICO format icon data.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None Input cannot be piped to this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.Icon

Returns an Icon object.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Add-PSSnapin Citrix.Common.Commands
C:\PS> $ctxIcon = Get-CtxIcon -FileName C:\Windows\System32\notepad.exe -index 0
C:\PS> $brokerIcon = New-BrokerIcon -EncodedIconData $ctxIcon.EncodedIconData
C:\PS> $desktopGroup = Get-BrokerDesktopGroup -Name 'MyDesktopGroup'
C:\PS> Set-BrokerDesktopGroup $desktopGroup -IconUid $brokerIcon.Uid
Extracts the first icon resource from notepad.exe, and sets this as the icon for a desktop group.
```

# New-BrokerMachine

Jan 04, 2017

Adds a machine that can be used to run desktops and applications.

## Syntax

```
New-BrokerMachine [-MachineName] <String> -CatalogUid <Int32> [-AssignedClientName <String>] [-AssignedIPAddress <String>] [-HostedMachineId <String>] [-HypervisorConnectionUid <Int32>] [-InMaintenanceMode <Boolean>] [-UUID <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

By adding a machine to a catalog, `New-BrokerMachine` adds a machine to the site, and is the first step in making the machine available to run users' desktops and applications. The machine may be physical or virtual.

For physical machines, you must specify the machine's SID and the catalog to which it will belong. For virtual machines which are not provisioned by MCS, you must also provide the hypervisor connection responsible for running the machine and the hosted machine ID by which the hypervisor recognizes the machine.

The machine must support the expected capabilities of the catalog: the catalog specifies a `SessionType` and a `MinimalFunctionalLevel`. The session support of the machine is determined by the type of Citrix VDA software installed (server or workstation) and the functional level depends on the version of the Citrix VDA software installed. The `New-BrokerMachine` command will complete successfully if these are not correct but the machine will be unable to register.

For more information about machines, see `about_Broker_Machines`.

## Related topics

[Add-BrokerMachine](#)

## Parameters

**-MachineName**<String>

Specify the name of the machine to create (in the form 'domain\machine'). A SID can also be specified.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-CatalogUid**<Int32>

The catalog to which this machine will belong.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-AssignedClientName**<String>

The client name to which this machine will be assigned. Machines can be assigned to multiple users, a single client IP address, or a single client name, but only to one of these categories at a time.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-AssignedIPAddress**<String>



The client IP address to which this machine will be assigned. Machines can be assigned to multiple users, a single client IP address, or a single client name, but only to one of these categories at a time.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-HostedMachineId**<String>

The unique ID by which the hypervisor recognizes the machine. Omit this for physical machines or MCS-provisioned VMs.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-HypervisorConnectionUid**<Int32>

The hypervisor connection that runs the machine. Omit this for physical machines or MCS-provisioned VMs.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-InMaintenanceMode**<Boolean>

Specifies whether the machine is initially in maintenance mode. A machine in maintenance mode is not available for new sessions, and for managed machines all automatic power management is disabled.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-UUID**<Guid>

An optional GUID for this machine.

|                        |                                              |
|------------------------|----------------------------------------------|
| Required?              | false                                        |
| Default Value          | A new GUID is generated if none is supplied. |
| Accept Pipeline Input? | true (ByPropertyName)                        |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.Machine

New-BrokerMachine returns the created machine.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> New-BrokerMachine -CatalogUid 2 -MachineName 'domain\machine'

This adds the physical machine with the specified SAM name to this site and places it in the specified catalog.

----- **EXAMPLE 2** -----

C:\PS> New-BrokerMachine -CatalogUid 2 -MachineName 'S-1-5-12-1234567890-1234567890-1234567890-1234'

This adds the physical machine with the specified SID to this site and places it in the specified catalog.

----- **EXAMPLE 3** -----

C:\PS> New-BrokerMachine -CatalogUid 2 -MachineName 'domain\machine' -HostedMachineId 'F8143B4F-7371-4efa-868A-54787EF9F64E' -HypervisorConnectionUid 5

This adds the virtual machine, running on the specified hypervisor, to this site and places it in the catalog.

----- **EXAMPLE 4** -----

C:\PS> \$m = New-BrokerMachine -CatalogUid 2 -MachineName 'domain\machine'

C:\PS> Add-BrokerMachine -InputObject \$m -DesktopGroup 3

This adds the specified physical machine to the site and uses Add-BrokerMachine to add it to a desktop group.

# New-BrokerMachineCommand

Jan 04, 2017

Creates a new command to deliver to a desktop.

## Syntax

```
New-BrokerMachineCommand -User <String> -Category <String> -CommandName <String> [-DesktopGroups <DesktopGroup[]>] [-SendTrigger <MachineCommandTrigger>] [-SendDeadline <TimeSpan>] [-CommandData <Byte[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-BrokerMachineCommand -SessionUid <Int64> -Category <String> -CommandName <String> [-SendTrigger <MachineCommandTrigger>] [-SendDeadline <TimeSpan>] [-CommandData <Byte[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-BrokerMachineCommand -MachineUid <Int32> -Category <String> -CommandName <String> [-SendTrigger <MachineCommandTrigger>] [-SendDeadline <TimeSpan>] [-CommandData <Byte[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-BrokerMachineCommand -Synchronous -MachineUid <Int32> -Category <String> -CommandName <String> [-CommandData <Byte[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Create a new command queued for delivery to a desktop. Commands are sent to a specific handler installed on the desktop using the Category parameter. Each handler has its own list of commands identified by the CommandName parameter. Optional command data can be provided using the CommandData parameter in a format specified by the handler.

Commands are targeted at a specific user, session or machine. Commands targeted at a user can be further be restricted to one or more desktop groups.

The SendTrigger is used to restrict the command to a specific event related to the target. For example, when the target machine registers or when the target user reconnects to a session. The command will be sent to the machine when the SendTrigger occurs for the target.

If the Synchronous switch is provided, the target must be a machine and no SendTrigger can be specified. The command is sent immediately to the machine if it is currently registered and fails if the machine is not registered.

Note that the combined length of the Category and CommandName is limited to 64 characters. The Category and CommandName must both be entirely alphanumeric and not include any white space.

## Related topics

[Get-BrokerMachineCommand](#)

[Remove-BrokerMachineCommand](#)

## Parameters

**-Category**<String>

The service on the desktop to send the command to.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-CommandName**<String>

The name of the command to send (as defined by the service).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-User**<String>

User whose desktop or session should receive the command.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | Any user.             |
| Accept Pipeline Input? | true (ByPropertyName) |

**-SessionUid**<Int64>

Currently logged on user session that should receive the command.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | Any session.          |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MachineUid**<Int32>

Specific machine that should receive the command.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Synchronous**<SwitchParameter>

Send the command immediately and block while waiting for the reply.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |       |
|------------------------|-------|
| Default Value          | false |
| Accept Pipeline Input? | false |

**-CommandData**<Byte[]>

Optional additional data to include with the command.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | None                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**-DesktopGroups**<DesktopGroup[]>

Further restrict the command targeted at a user to machines in these desktop groups.

|                        |                                  |
|------------------------|----------------------------------|
| Required?              | false                            |
| Default Value          | No restriction by desktop group. |
| Accept Pipeline Input? | true (ByPropertyName)            |

### -SendTrigger<MachineCommandTrigger>

Queue command for delivery until this particular event occurs. Valid values are NextContact, Broker, LogOn, Logoff, Disconnect and Reconnect.

|                        |                                                                                                       |
|------------------------|-------------------------------------------------------------------------------------------------------|
| Required?              | false                                                                                                 |
| Default Value          | Default value is 'NextContact' so the command is sent during the next communication with the desktop. |
| Accept Pipeline Input? | true (ByPropertyName)                                                                                 |

### -SendDeadline<TimeSpan>

Automatically cancel the command if it not delivered before the specified time span passes.

|                        |                                 |
|------------------------|---------------------------------|
| Required?              | false                           |
| Default Value          | Command expires after 24 hours. |
| Accept Pipeline Input? | true (ByPropertyName)           |

### Input Type

None No parameter is accepted from the input pipeline.

### Return Values

Citrix.Broker.Admin.SDK.MachineCommand

New command that was added to the command queue.Citrix.Broker.Admin.SDK.MachineSynchronousCommandResponse

When the Synchronous option is used, the command is immediately sent to the specified machine and processed. The SDK object returned describes the command and the result of this command processing.

### Notes

Commands are subject to delegated administration restrictions based on the desktop group, category and command name.

### Examples

#### ----- EXAMPLE 1 -----

New-BrokerMachineCommand -Category "UPM" -CommandName "ResetProfile" -DesktopGroups 1 -UserUid 23 -SendTrigger Authentication  
Instruct the User Profile Manager service to execute the "ResetProfile" command when user 23 logs on to any machine in desktop group 1

#### ----- EXAMPLE 2 -----

New-BrokerMachineCommand -Synchronous -Category "MonitorService" -CommandName "EnableLogging" -MachineUid 13  
Instruct the monitor service to immediately execute the "EnableLogging" command on the machine having Uid 13.

# New-BrokerMachineConfiguration

Jan 04, 2017

Creates a new machine configuration associated with an existing configuration slot.

## Syntax

```
New-BrokerMachineConfiguration -ConfigurationSlotUid <Int32> -LeafName <String> -Policy <Byte[]> [-Description <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

## Detailed Description

Creates a new machine configuration containing settings that match the SettingsGroup of the associated configuration slot. This machine configuration can then be applied to a desktop group to have the settings applied to machines in that group.

The SettingsGroup of the configuration slot restricts the permitted settings. Use the SDK snap-in that matches the SettingsGroup to create the encoded settings data.

## Related topics

[Get-BrokerMachineConfiguration](#)

[Set-BrokerMachineConfiguration](#)

[Rename-BrokerMachineConfiguration](#)

[Remove-BrokerMachineConfiguration](#)

[Add-BrokerMachineConfiguration](#)

## Parameters

**-ConfigurationSlotUid**<Int32>

Unique identifier of the configuration slot to associate with this machine configuration.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | None                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LeafName**<String>

Name of the new machine configuration. This must be unique amongst the machine configurations associated with the same configuration slot.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | None                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Policy**<Byte[]>

A binary array of encoded settings (policy) data created with the SDK snap-in that matches the SettingsGroup of the configuration slot.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | None                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Description**<String>

Description of the new machine configuration.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.MachineConfiguration

New-BrokerMachineConfiguration returns the newly created configuration

**Notes**

Delegated Administration can be used to restrict the configuration slots that an administrator can use and hence which components of the system that can be configured.

**Examples**

----- **EXAMPLE 1** -----

New-BrokerMachineConfiguration -LeafName "Finance Department" -Description "Finance Dept. User Profile Management policy" -Policy \$policy -ConfigurationSlotUid \$csUid  
 Creates a new configuration named "%SlotName%\Finance Department" where %SlotName% is the name of the configuration slot having the Uid \$csUid. The encoded settings in the \$policy variable must match the SettingsGroup of the configuration slot having the Uid \$csUid.



# New-BrokerPowerTimeScheme

Jan 04, 2017

Creates a new power time scheme for a desktop group.

## Syntax

```
New-BrokerPowerTimeScheme [-Name] <String> -DaysOfWeek <TimeSchemeDays> -DesktopGroupUid <Int32> [-DisplayName <String>] [-PeakHours <Boolean[]>] [-PoolSize <Int32[]>] [-PoolUsingPercentage <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The New-BrokerPowerTimeScheme cmdlet adds a new power time scheme to be associated with a desktop group. The power time scheme must relate to days of the week that are not already covered by an existing power time scheme.

Each power time scheme is associated with a particular desktop group, and covers one or more days of the week, defining which hours of those days are considered peak times and which are off-peak times. In addition, the time scheme defines a pool size value for each hour of the day for the days of the week covered by the time scheme. No one desktop group can be associated with two or more time schemes that cover the same day of the week.

See 'help about\_Broker\_PowerManagement' for a detailed description of the power policy mechanism and pool size management.

## Related topics

[Get-BrokerPowerTimeScheme](#)

[Set-BrokerPowerTimeScheme](#)

[Remove-BrokerPowerTimeScheme](#)

[Rename-BrokerPowerTimeScheme](#)

## Parameters

**-Name**<String>

Specifies the administrative name of the new power time scheme. Each scheme must have a name which is unique within the site.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DaysOfWeek**<TimeSchemeDays>

Specifies the pattern of days of the week that the power time scheme covers.

Valid values are (singly or a list of) Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, Weekdays and Weekend.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopGroupUid**<Int32>

Specifies the desktop group that the power time scheme applies to.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DisplayName**<String>

Specifies the name of the new power time scheme as displayed in the DesktopStudio console. Each scheme associated with a desktop group must have a display name which is unique within its desktop group, although the same display name can be used on power schemes for different desktop groups.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PeakHours**<Boolean[]>

A set of 24 boolean flag values, one for each hour of the day. The first value in the array relates to midnight to 00:59, the next one to 1 AM to 01:59 and so on, with the last array element relating to 11 PM to 11:59. If the flag is \$true it means that the associated hour of the day is considered a peak time; if \$false it means that it is considered off-peak.

|                        |                                                   |
|------------------------|---------------------------------------------------|
| Required?              | false                                             |
| Default Value          | 24 \$false values, meaning all hours are off-peak |
| Accept Pipeline Input? | true (ByPropertyName)                             |

**-PoolSize**<Int32[]>

A set of 24 integer values, one for each hour of the day. The first value in the array relates to midnight to 00:59, the next one to 1 AM to 01:59 and so on, with the last array element relating to 11 PM to 11:59. The value defines the number of machines (either as an absolute number or a percentage of the machines in the desktop group) that are to be maintained in a running state, whether they are in use or not. A value of -1 has special meaning: pool size management does not apply during such hours.

|                        |                                                                       |
|------------------------|-----------------------------------------------------------------------|
| Required?              | false                                                                 |
| Default Value          | 24 values of '-1', meaning no pool size management is to be performed |
| Accept Pipeline Input? | true (ByPropertyName)                                                 |

**-PoolUsingPercentage**<Boolean>

A boolean flag to indicate whether the integer values in the pool size array are to be treated as absolute values (if this value is \$false) or as percentages of the number of machines in the desktop group (if this value is \$true).

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

#### Input Type

None You cannot pipe input into this cmdlet.

#### Return Values

Citrix.Broker.Admin.SDK.PowerTimeScheme

New-BrokerPowerTimeScheme returns the newly created power time scheme.

#### Examples

----- **EXAMPLE 1** -----

```
C:\PS> New-BrokerPowerTimeScheme -Name 'First Half Week' -DaysOfWeek Weekend,Monday,Tuesday -DesktopGroupUid 3 -PeakHours (0..23 | %{ $_ -gt 8 -and $_ -lt 18 } )
```

Creates a new scheme attached to the desktop group whose UID value is 3. This new scheme covers the weekend and Monday and Tuesday, and defines 'peak' hours as 9am to 17:59, with all other times being 'off-peak'. No pool size values are supplied, so all size values for all the hours default to -1.

# New-BrokerRebootSchedule

Jan 04, 2017

Creates a new reboot schedule for a desktop group.

## Syntax

```
New-BrokerRebootSchedule [-DesktopGroupName <String> -RebootDuration <Int32> [-Day <RebootScheduleDays>] [-Enabled <Boolean>] [-Frequency <RebootScheduleFrequency>] [-StartTime <TimeSpan>] [-WarningDuration <Int32>] [-WarningMessage <String>] [-WarningTitle <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-BrokerRebootSchedule -DesktopGroupUid <Int32> -RebootDuration <Int32> [-Day <RebootScheduleDays>] [-Enabled <Boolean>] [-Frequency <RebootScheduleFrequency>] [-StartTime <TimeSpan>] [-WarningDuration <Int32>] [-WarningMessage <String>] [-WarningTitle <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The New-BrokerRebootSchedule cmdlet is used to define a reboot schedule for a desktop group.

## Related topics

[Get-BrokerRebootSchedule](#)

[Set-BrokerRebootSchedule](#)

[Remove-BrokerRebootSchedule](#)

[Start-BrokerRebootCycle](#)

## Parameters

**-DesktopGroupName**<String>

The name of the desktop group that this reboot schedule is applied to.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-RebootDuration**<Int32>

Approximate maximum number of minutes over which the scheduled reboot cycle runs.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopGroupUid**<Int32>

The Uid of the desktop group that this reboot schedule is applied to.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Day**<RebootScheduleDays>

For weekly schedules, the day of the week on which the scheduled reboot-cycle starts (one of Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday).

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Enabled**<Boolean>

Boolean that indicates if the new reboot schedule is enabled.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Frequency**<RebootScheduleFrequency>

Frequency with which this schedule runs (either Weekly or Daily).

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-StartTime**<TimeSpan>

Time of day at which the scheduled reboot cycle starts (HH:MM).

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-WarningDuration**<Int32>

Time prior to the initiation of a machine reboot at which warning message is displayed in all user sessions on that machine. If the warning duration is zero then no message is displayed.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-WarningMessage**<String>

Warning message displayed in user sessions on a machine scheduled for reboot. If the message is blank then no message is displayed.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-WarningTitle**<String>

The window title used when showing the warning message in user sessions on a machine scheduled for reboot.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None Input cannot be piped to this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.RebootSchedule

Examples

----- **EXAMPLE 1** -----

```
C:\PS> New-BrokerRebootSchedule -DesktopGroupName BankTellers -Frequency Daily -StartTime "02:00" -Enabled $true -Duration 120
```

Schedules the machines in the desktop group named 'BankTellers' to be rebooted every night between 2 AM and 4 AM.

----- **EXAMPLE 2** -----

```
C:\PS> New-BrokerRebootSchedule -DesktopGroupUid 17 -Frequency Weekly -Day Saturday -StartTime "01:00" -Enabled $true -Duration 240 -WarningTitle "WARNING: Reboot pending" -WarningMes
```

Schedules the machines in the desktop group having Uid 17 to be rebooted every Saturday night between 1 AM and 5 AM. Ten minutes prior to rebooting, each machine will display a message box with the title "WARNING: Reboot pending" and message "Save your work" in every user session.

# New-BrokerRemotePCAccount

Jan 04, 2017

Create a new RemotePCAccount.

## Syntax

```
New-BrokerRemotePCAccount -CatalogUid <Int32> -OU <String> [-AllowSubfolderMatches <Boolean>] [-MachinesExcluded <String[]>] [-MachinesIncluded <String[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Create a new RemotePCAccount. A RemotePCAccount defines machine filters to support Remote PC automation adding unconfigured machines to catalogs.

## Related topics

[Get-BrokerRemotePCAccount](#)

[Set-BrokerRemotePCAccount](#)

[Remove-BrokerRemotePCAccount](#)

## Parameters

**-CatalogUid**<Int32>

Specifies the catalog which Remote PC automation adds an unconfigured machine to if it matches this RemotePCAccount.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-OU**<String>

Specifies the DN of an AD container, or has the special value 'any'.

When an AD container is specified a machine may only match with the RemotePCAccount when the AD computer object is located relative to the OU.

When 'any' is specified the location of the AD computer object is ignored for purposes of matching this RemotePCAccount. The machine must still meet the MachinesIncluded and MachinesExcluded filters for a match to occur.

In the event that a machine matches with multiple RemotePCAccounts then the RemotePCAccount OU with the longest canonical name takes precedence. The special 'any' OU is treated as lowest priority.

Note that the OU value of every RemotePCAccount must be unique, and this includes only one 'any' entry being permitted.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-AllowSubfolderMatches**<Boolean>

When true a machine matches this RemotePCAccount if the AD computer object exists within the container specified by the OU property, or within a child container of the OU.

When false the AD computer object only matches if it exists directly in the AD container specified by the OU property.

This property is not meaningful when OU has the special value 'any'.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MachinesExcluded**<String[]>

MachinesExcluded specifies a set of strings that can include asterisk wildcards. If a machine name matches any entries in MachinesExcluded then it cannot match with this RemotePCAccount regardless of whether there is a MachinesIncluded match.

Matches are performed against the domain name joined with the machine name by a backslash (DOMAIN\MACHINE), e.g.:

DOMAIN1\M\*

DOMAIN\*\M\*

\*\M\*

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | @()                   |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MachinesIncluded**<String[]>

MachinesIncluded specifies a set of strings that can include asterisk wildcards. A machine may only match with this RemotePCAccount if it matches a MachinesIncluded entry and does not match any MachinesExcluded entries.

Matches are performed against the domain name joined with the machine name by a backslash (DOMAIN\MACHINE), e.g.:

DOMAIN1\M\*

DOMAIN\*\M\*

\*\M\*

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | @('*')                |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and



Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.RemotePCAccount

The newly created RemotePCAccount.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> New-BrokerRemotePCAccount -OU 'ou=MyOU,dc=MyDomain,dc=com' -CatalogUid 42  
Create a RemotePCAccount that adds unconfigured machines with computer objects in MyOU, into catalog 42.
```

----- **EXAMPLE 2** -----

```
C:\PS> New-BrokerRemotePCAccount -OU 'any' -CatalogUid 42 -MachinesIncluded @('DOMAIN1\*') -MachinesExcluded @('DOMAIN1\*JOHNDOE*')  
Create a RemotePCAccount matching unconfigured machines from DOMAIN1, except those with hostnames containing JOHNDOE, and add them to catalog 42.
```

----- **EXAMPLE 3** -----

```
C:\PS> New-BrokerRemotePCAccount -OU 'any' -CatalogUid 42  
Create a RemotePCAccount that matches any unconfigured machine, causing automation to add matching machines to catalog 42.
```

# New-BrokerSessionLinger

Jan 04, 2017

Creates a new session linger setting for a desktop group.

## Syntax

```
New-BrokerSessionLinger [-DesktopGroupName] <String> [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-BrokerSessionLinger -DesktopGroupUid <Int32> [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The New-BrokerSessionLinger cmdlet is used to define a session linger setting for a desktop group.

Note that each desktop group can only have a single session linger setting. Session lingering only applies to application sessions.

## Related topics

[Get-BrokerSessionLinger](#)

[Set-BrokerSessionLinger](#)

[Remove-BrokerSessionLinger](#)

## Parameters

**-DesktopGroupName**<String>

The name of the desktop group that this linger setting is applied to.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopGroupUid**<Int32>

The Uid of the desktop group that this linger setting is applied to.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Enabled**<Boolean>

Boolean that indicates if the new session linger is enabled.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MaxAverageLoadThreshold**<Int32>

Specifies the average load threshold across the desktop group. When the threshold hits, lingering sessions across the group be terminated to reduce load. Sessions that have been lingering the longest will be chosen first.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 0                     |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MaxLoadPerMachineThreshold**<Int32>

Specifies the maximum load threshold per machine in the desktop group. When the threshold hits, lingering sessions on each loaded machine will be terminated to reduce load. Sessions that have been lingering the longest will be chosen first.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 0                     |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MaxTimeBeforeDisconnect**<TimeSpan>

Specifies the time by which a lingering session will be disconnected. The disconnect timer is optional, but when specified the terminate timer needs to be also set. The disconnect time cannot be greater than the terminate time. When the disconnect and terminate times are the same, the terminate timer takes precedence. The disconnect timer needs to be paired with a session termination condition like the terminate timer or one of load threshold settings.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 15 minutes            |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MaxTimeBeforeTerminate**<TimeSpan>

Specifies the time by which a lingering session will be terminated. The terminate timer is not optional when timers are configured. When the disconnect and terminate times are the same, the terminate timer takes precedence.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          | 8 hours               |
| Accept Pipeline Input? | true (ByPropertyName) |

**-UserFilterEnabled**<Boolean>

Specifies whether the session linger's user filter is enabled or disabled. Where the user filter is enabled, lingering is enabled only to users who appear in the filter (either explicitly or by virtue of group membership).

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Depends on parameter Parameters can be piped by property name.

**Return Values**

Citrix.Broker.Admin.SDK.SessionLinger

New-BrokerSessionLinger returns a session linger object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> New-BrokerSessionLinger -DesktopGroupName test -Enabled $true -MaxTimeBeforeDisconnect 0:30 -MaxTimeBeforeTerminate 1:00
```

Creates a new session linger setting with a disconnect timer of 30 minutes and terminate timer of 1 hour.

# New-BrokerSessionPreLaunch

Jan 04, 2017

Creates a new session pre-launch setting for a desktop group.

## Syntax

```
New-BrokerSessionPreLaunch [-DesktopGroupName] <String> [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-BrokerSessionPreLaunch -DesktopGroupUid <Int32> [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The New-BrokerSessionPreLaunch cmdlet is used to define a session pre-launch setting for a desktop group.

Note that each desktop group can only have a single session pre-launch setting. Session pre-launch only applies to application sessions.

## Related topics

[Get-BrokerSessionPreLaunch](#)

[Set-BrokerSessionPreLaunch](#)

[Remove-BrokerSessionPreLaunch](#)

## Parameters

**-DesktopGroupName**<String>

The name of the desktop group that this pre-launch setting is applied to.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopGroupUid**<Int32>

The Uid of the desktop group that this pre-launch setting is applied to.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Enabled**<Boolean>

Boolean that indicates if the new session pre-launch is enabled.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | true                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MaxAverageLoadThreshold**<Int32>

Specifies the average load threshold across the desktop group. When the threshold hits, pre-launched sessions across the group be terminated to reduce load. Sessions that have been pre-launched the longest will be chosen first.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 0                     |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MaxLoadPerMachineThreshold**<Int32>

Specifies the maximum load threshold per machine in the desktop group. When the threshold hits, pre-launched sessions on each loaded machine will be terminated to reduce load. Sessions that have been pre-launched the longest will be chosen first.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 0                     |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MaxTimeBeforeDisconnect**<TimeSpan>

Specifies the time by which a pre-launched session will be disconnected. The disconnect timer is optional, but when specified the terminate timer needs to be also set. The disconnect time cannot be greater than the terminate time. When the disconnect and terminate times are the same, the terminate timer takes precedence. The disconnect timer needs to be paired with a session termination condition like the terminate timer or one of load threshold settings.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | 15 minutes            |
| Accept Pipeline Input? | true (ByPropertyName) |

**-MaxTimeBeforeTerminate**<TimeSpan>

Specifies the time by which a pre-launched session will be terminated. The terminate timer is not optional when timers are configured. When the disconnect and terminate times are the same, the terminate timer takes precedence.

|               |         |
|---------------|---------|
| Required?     | false   |
| Default Value | 8 hours |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-UserFilterEnabled**<Boolean>

Specifies whether the session pre-launch's user filter is enabled or disabled. Where the user filter is enabled, pre-launch is enabled only to users who appear in the filter (either explicitly or by virtue of group membership).

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | false                 |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Depends on parameter Parameters can be piped by property name.

**Return Values**

Citrix.Broker.Admin.SDK.SessionPreLaunch

New-BrokerSessionPreLaunch returns a session pre-launch object.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> New-BrokerSessionPreLaunch -DesktopGroupName test -Enabled \$true -MaxTimeBeforeDisconnect 0:30 -MaxTimeBeforeTerminate 1:00  
Creates a new session pre-launch setting with a disconnect timer of 30 minutes and terminate timer of 1 hour.





# New-BrokerTag

Jan 04, 2017

Creates a new tag.

## Syntax

```
New-BrokerTag [-Name] <String> [-UUID <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Creates a tag that can be associated with other objects using Add-BrokerTag.

## Related topics

[Add-BrokerTag](#)

[Get-BrokerTag](#)

[Remove-BrokerTag](#)

[Rename-BrokerTag](#)

## Parameters

**-Name**<String>

Specifies a name for the new tag.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-UUID**<Guid>

Specifies a UUID for the new tag. When not specified, a UUID is automatically assigned.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None Input cannot be piped to this cmdlet.

**Return Values**

Citrix.Broker.Admin.SDK.Tag

Outputs the generated tag.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> New-BrokerTag -Name 'Tag1'
```

Creates a new tag with name 'Tag1'.

----- **EXAMPLE 2** -----

```
C:\PS> New-BrokerTag 'Tag2' | Add-BrokerTag -Desktop $desktop
```

Creates a new tag with name 'Tag2' and associates it with Desktop \$desktop.

# New-BrokerUser

Jan 04, 2017

Creates a new broker user object

## Syntax

```
New-BrokerUser [-SID] <SecurityIdentifier> [-AdminAddress <String>] [<CommonParameters>]
```

```
New-BrokerUser [-Name] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The New-BrokerUser cmdlet creates a new broker object to represent a user identity (or the identity of a group of users). The object is created local to the PowerShell environment in which the cmdlet is run; no new user object is created in the broker configuration, unless the object is added to another broker object, such as a machine or a desktop. For details, see Add-BrokerUser.

The identity of the user or group must be specified using either the Name or SID parameter

## Related topics

[Add-BrokerUser](#)

[Get-BrokerUser](#)

[Remove-BrokerUser](#)

## Parameters

**-SID**<SecurityIdentifier>

The SID of the user or group

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          | null  |
| Accept Pipeline Input? | false |

**-Name**<String>

The name of the user or group

|               |      |
|---------------|------|
| Required?     | true |
| Default Value | null |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.Broker.Admin.SDK.User

The broker user object

Notes

Typically, broker user objects are created implicitly using the Add-BrokerUser cmdlet with a user name or SID.

Examples

----- **EXAMPLE 1** -----

```
$user = New-BrokerUser DOMAIN\UserName
Create a broker user object for the specified user.
```

----- **EXAMPLE 2** -----

```
$user = New-BrokerUser -SID S-1-5-23-1763203430-193137401-908696819-3450
Create a broker user object for the specified user.
```

# Remove-BrokerAccessPolicyRule

Jan 04, 2017

Deletes a rule from the site's access policy.

## Syntax

```
Remove-BrokerAccessPolicyRule [-InputObject] <AccessPolicyRule[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerAccessPolicyRule [-Name] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerAccessPolicyRule cmdlet deletes a rule from the site's access policy.

An access policy rule defines a set of connection filters and access control rights relating to a desktop group. These allow fine-grained control of what access is granted to a desktop group based on details of, for example, a user's endpoint device, its address, and the user's identity.

Deleting a rule does not affect existing user sessions, but it may result in users being unable to launch new sessions, or reconnect to disconnected sessions if access to the desktop group delivering those sessions was granted by the deleted rule.

## Related topics

[New-BrokerAccessPolicyRule](#)

[Get-BrokerAccessPolicyRule](#)

[Set-BrokerAccessPolicyRule](#)

[Rename-BrokerAccessPolicyRule](#)

## Parameters

**-InputObject** <AccessPolicyRule[]>

The access policy rule to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name** <String>

The name of the access policy rule to be deleted.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

#### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

#### Input Type

Citrix.Broker.Admin.SDK.AccessPolicyRule The access policy rule to be deleted.

#### Return Values

None

#### Examples

##### ----- EXAMPLE 1 -----

```
C:\PS> Remove-BrokerAccessPolicyRule 'Temp Staff'
```

Deletes the access policy rule called Temp Staff. Existing sessions are not affected, but if access was granted by the deleted rule users may be unable to reconnect to sessions if they are subsequently disconnected.

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerAccessPolicyRule -IncludedUsers sales\johndoe | Remove-BrokerAccessPolicyRule
```

Deletes all access policy rules explicitly granting user SALES\johndoe access to any desktop group in the site. Any existing desktop sessions for the user are not affected. The user may still be able to access site resources by access policy rules that grant access through group membership or non-user-based connection filters.



# Remove-BrokerAccessPolicyRuleMetadata

Jan 04, 2017

Deletes AccessPolicyRule Metadata from the AccessPolicyRule objects

## Syntax

```
Remove-BrokerAccessPolicyRuleMetadata [-InputObject] <AccessPolicyRule[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerAccessPolicyRuleMetadata cmdlet deletes Metadata from the AccessPolicyRule objects.

## Related topics

## Parameters

**-InputObject**<AccessPolicyRule[]>

Specifies the AccessPolicyRule object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerAccessPolicyRule You can pipe the AccessPolicyRule to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerAccessPolicyRuleMetadata -InputObject \$obj-Uid -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for the AccessPolicyRule whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerAccessPolicyRule | Remove-BrokerAccessPolicyRuleMetadata -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for all the AccessPolicyRule in the site

# Remove-BrokerAdminFolder

Jan 04, 2017

Removes an admin folder.

## Syntax

```
Remove-BrokerAdminFolder [-InputObject] <AdminFolder[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-BrokerAdminFolder [-Name] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerAdminFolder cmdlet removes an existing admin folder.

Remove-BrokerAdminFolder will not remove a folder if it contains any other objects (e.g. sub-folders or applications).

## Related topics

[Get-BrokerAdminFolder](#)

[New-BrokerAdminFolder](#)

## Parameters

**-InputObject**<AdminFolder[]>

Identifies the folder to remove

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The name pattern of folder(s) to remove

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.AdminFolder The admin folder objects can be specified as input.

**Return Values**

None

This cmdlet does not return any output.

**Examples**

----- **EXAMPLE 1** -----

```
Remove-BrokerAdminFolder F1\F2\
Removes the folder called F2 within the folder F1\
```

# Remove-BrokerAdminFolderMetadata

Jan 04, 2017

Deletes AdminFolder Metadata from the AdminFolder objects

## Syntax

```
Remove-BrokerAdminFolderMetadata [-InputObject] <AdminFolder[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerAdminFolderMetadata cmdlet deletes Metadata from the AdminFolder objects.

## Related topics

## Parameters

**-InputObject**<AdminFolder[]>

Specifies the AdminFolder object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerAdminFolder You can pipe the AdminFolder to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerAdminFolderMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the AdminFolder whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerAdminFolder | Remove-BrokerAdminFolderMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the AdminFolder in the site

# Remove-BrokerAppAssignmentPolicyRule

Jan 04, 2017

Deletes an application rule from the site's assignment policy.

## Syntax

```
Remove-BrokerAppAssignmentPolicyRule [-InputObject] <AppAssignmentPolicyRule[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerAppAssignmentPolicyRule [-Name] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerAppAssignmentPolicyRule cmdlet deletes an application rule from the site's assignment policy.

An application rule in the assignment policy defines the users who are entitled to a self-service persistent machine assignment from the rule's desktop group; once assigned the machine can run one or more applications published from the group.

Deleting an application rule does not remove machine assignments that have already been made by the rule, nor does it affect active sessions to those machines in any way.

## Related topics

[New-BrokerAppAssignmentPolicyRule](#)

[Get-BrokerAppAssignmentPolicyRule](#)

[Set-BrokerAppAssignmentPolicyRule](#)

[Rename-BrokerAppAssignmentPolicyRule](#)

## Parameters

**-InputObject** <AppAssignmentPolicyRule[]>

The application rule to be deleted from the assignment policy.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name** <String>

The name of the application rule to be deleted from the assignment policy.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.AppAssignmentPolicyRule The application rule to be deleted from the assignment policy.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerAppAssignmentPolicyRule 'Temp Staff'
```

Deletes the application rule called Temp Staff from the assignment policy. Access to machines already assigned by this rule is not affected in any way.



----- **EXAMPLE 2** -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Sales Support'
```

```
C:\PS> Get-BrokerAppAssignmentPolicyRule -DesktopGroupUid $dg.Uid | Remove-BrokerAppAssignmentPolicyRule
```

Deletes the application rule for the Sales Support desktop group from the site's assignment policy. This prevents any further machine assignments being made from this group, but it does not affect existing assignments made by the rule.

# Remove-BrokerAppEntitlementPolicyRule

Jan 04, 2017

Deletes an application rule from the site's entitlement policy.

## Syntax

```
Remove-BrokerAppEntitlementPolicyRule [-InputObject] <AppEntitlementPolicyRule[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerAppEntitlementPolicyRule [-Name] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerAppEntitlementPolicyRule cmdlet deletes an application rule from the site's entitlement policy.

An application rule in the entitlement policy defines the users who are allowed per-session access to a machine to run one or more applications published from the rule's desktop group.

Deleting a rule does not affect existing sessions launched using the rule, but users cannot reconnect to those sessions if they are subsequently disconnected.

## Related topics

[New-BrokerAppEntitlementPolicyRule](#)

[Get-BrokerAppEntitlementPolicyRule](#)

[Set-BrokerAppEntitlementPolicyRule](#)

[Rename-BrokerAppEntitlementPolicyRule](#)

## Parameters

**-InputObject**<AppEntitlementPolicyRule[]>

The application rule to be deleted from the entitlement policy.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The name of the application rule to be deleted from the entitlement policy.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule The application rule to be deleted from the entitlement policy.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerAppEntitlementPolicyRule 'Temp Workers'

Deletes the application rule called Temp Workers from the entitlement policy rule. Existing application sessions launched using that rule are not affected, but users cannot reconnect to those sessions if they are subsequently disconnected.

----- **EXAMPLE 2** -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Customer Support'
```

```
C:\PS> Get-BrokerAppEntitlementPolicyRule -DesktopGroupUid $dg.Uid | Remove-BrokerAppEntitlementPolicyRule
```

Deletes the application rule from the entitlement policy rule applied to the Customer Support desktop group. This effectively removes all access to the applications published from this group. Existing application sessions are not affected, but users cannot reconnect to those sessions if they are subsequently disconnected.

# Remove-BrokerApplication

Jan 04, 2017

Deletes one or more applications, or an association of an application.

## Syntax

```
Remove-BrokerApplication [-InputObject] <Application[]> [-Force] [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerApplication [-Name] <String> [-Force] [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerApplication cmdlet deletes one or more applications, or you can use it to delete just the association of an application to a desktop group.

Its usage dictates the behavior of the cmdlet. If only the application is specified as a parameter, then the cmdlet deletes the application. It also deletes any associations this application has with other objects, such as with access policy rules or desktop groups. More specifically, when an application is deleted the following happens:

- o The association to any desktop groups is removed.
- o The association to any tags is removed.
- o Any configured file-type association objects for this application are deleted.
- o The association to any user accounts is removed.
- o The association to any access session conditions is removed.
- o The access policy rule object for this application, if one existed, is deleted.
- o Finally, the application object itself is deleted.

Note that if the application is in use by a user then the application cannot be deleted.

If more than just the application is supplied as a parameter to the cmdlet (for instance, if a DesktopGroup object is also specified) then the application is not deleted. Instead, only the association from the application to that desktop group is removed.

## Related topics

[New-BrokerApplication](#)

[Add-BrokerApplication](#)

[Get-BrokerApplication](#)

[Rename-BrokerApplication](#)

[Move-BrokerApplication](#)

## Set-BrokerApplication

### Parameters

#### **-InputObject**<Application[]>

Specifies the applications to delete. The Uid can also be substituted for the application objects.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

#### **-Name**<String>

Specifies the name of the application to remove.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

#### **-Force**<SwitchParameter>

Remove application even if it's in use. Removing an application that is currently in use, can potentially leave an application session containing no applications. If all the applications that are currently active in a disconnected application session are removed, the user will be unable to reconnect to the session. Forcing removal of an in-use application does not impact the actual session itself.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

#### **-DesktopGroup**<DesktopGroup>

Specifies the desktop group that this application should no longer be associated with. The Uid or Name can also be substituted for the desktop group object.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                |
|------------------------|----------------|
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.Application The application objects can be specified as input.

**Return Values**

None

This cmdlet does not return any output.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerApplication "Notepad"

This command deletes the application that has a BrowserName of "Notepad".

----- **EXAMPLE 2** -----

```
C:\PS> $app = Get-BrokerApplication -BrowserName "Notepad"
```

```
C:\PS> $group = Get-BrokerDesktopGroup -Name "Private DesktopGroup"
```

```
C:\PS> Remove-BrokerApplication -InputObject $app -DesktopGroup $group
```

This command removes the association of the desktop group that has a name of "Private DesktopGroup" from the application that has a BrowserName of "Notepad". It does not otherwise modify the application.



# Remove-BrokerApplicationInstanceMetadata

Jan 04, 2017

Deletes ApplicationInstance Metadata from the ApplicationInstance objects

## Syntax

```
Remove-BrokerApplicationInstanceMetadata [-InputObject] <ApplicationInstance[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerApplicationInstanceMetadata cmdlet deletes Metadata from the ApplicationInstance objects.

## Related topics

## Parameters

**-InputObject**<ApplicationInstance[]>

Specifies the ApplicationInstance object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerApplicationInstance You can pipe the ApplicationInstance to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerApplicationInstanceMetadata -InputObject \$obj-Uid -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for the ApplicationInstance whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerApplicationInstance | Remove-BrokerApplicationInstanceMetadata -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for all the ApplicationInstance in the site

# Remove-BrokerApplicationMetadata

Jan 04, 2017

Deletes Application Metadata from the Application objects

## Syntax

```
Remove-BrokerApplicationMetadata [-InputObject] <Application[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerApplicationMetadata cmdlet deletes Metadata from the Application objects.

## Related topics

## Parameters

**-InputObject**<Application[]>

Specifies the Application object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerApplication You can pipe the Application to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerApplicationMetadata -InputObject \$obj-Uid -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for the Application whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerApplication | Remove-BrokerApplicationMetadata -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for all the Application in the site

# Remove-BrokerAssignmentPolicyRule

Jan 04, 2017

Deletes a desktop rule from the site's assignment policy.

## Syntax

```
Remove-BrokerAssignmentPolicyRule [-InputObject] <AssignmentPolicyRule[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerAssignmentPolicyRule [-Name] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerAssignmentPolicyRule cmdlet deletes a desktop rule from the site's assignment policy.

A desktop rule in the assignment policy defines the users who are entitled to self-service persistent machine assignments from the rule's desktop group. A rule defines how many machines a user is allowed from the group for delivery of full desktop sessions.

Deleting a desktop rule does not remove machine assignments that have already been made by the rule, nor does it affect active sessions to those machines in any way.

## Related topics

[New-BrokerAssignmentPolicyRule](#)

[Get-BrokerAssignmentPolicyRule](#)

[Set-BrokerAssignmentPolicyRule](#)

[Rename-BrokerAssignmentPolicyRule](#)

## Parameters

**-InputObject** <AssignmentPolicyRule[]>

The desktop rule to be deleted from the assignment policy.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name** <String>

The name of the desktop rule to be deleted from the assignment policy.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.AssignmentPolicyRule The desktop rule to be deleted from the assignment policy.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerAssignmentPolicyRule 'Temp Staff'

Deletes the desktop rule called Temp Staff from the assignment policy. Access to machines already assigned by this rule is not affected in any way.

----- **EXAMPLE 2** -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Sales Support'
```

```
C:\PS> Get-BrokerAssignmentPolicyRule -DesktopGroupUid $dg.Uid | Remove-BrokerAssignmentPolicyRule
```

Deletes all desktop rules for the Sales Support desktop group from the site's assignment policy. This prevents any further machine assignments being made from this group, but it does not affect existing assignments made by these rules.

# Remove-BrokerAssignmentPolicyRuleMetadata

Jan 04, 2017

Deletes AssignmentPolicyRule Metadata from the AssignmentPolicyRule objects

## Syntax

```
Remove-BrokerAssignmentPolicyRuleMetadata [-InputObject] <AssignmentPolicyRule[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerAssignmentPolicyRuleMetadata cmdlet deletes Metadata from the AssignmentPolicyRule objects.

## Related topics

## Parameters

**-InputObject**<AssignmentPolicyRule[]>

Specifies the AssignmentPolicyRule object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerAssignmentPolicyRule You can pipe the AssignmentPolicyRule to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerAssignmentPolicyRuleMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the AssignmentPolicyRule whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerAssignmentPolicyRule | Remove-BrokerAssignmentPolicyRuleMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the AssignmentPolicyRule in the site

# Remove-BrokerCatalog

Jan 04, 2017

Removes catalogs from the site.

## Syntax

```
Remove-BrokerCatalog [-InputObject] <Catalog[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerCatalog [-Name] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Remove catalogs from the site.

In order to remove a catalog from a site, the catalog must not contain machines. To remove a machine from a catalog use the Remove-BrokerMachine cmdlet. Note: in order to remove a machine from a catalog, it must not belong to a desktop group.

## Related topics

[New-BrokerCatalog](#)

[Get-BrokerCatalog](#)

[Rename-BrokerCatalog](#)

[Set-BrokerCatalog](#)

[New-BrokerDesktopGroup](#)

[Remove-BrokerDesktopGroup](#)

## Parameters

**-InputObject**<Catalog[]>

Specifies the catalog objects to delete.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the catalog to delete.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.Catalog You can pipe the catalogs to be deleted to Remove-BrokerCatalog.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerCatalog -Name "MyCatalog"
C:\PS> Remove-BrokerCatalog -InputObject (Get-BrokerCatalog -Name "MyCatalog")
```

These commands delete the catalog with the name "MyCatalog".

----- **EXAMPLE 2** -----

```
C:\PS> Remove-BrokerCatalog -Name 'test'
```

This command deletes all catalogs with names beginning with "test".

----- **EXAMPLE 3** -----

```
C:\PS> Get-BrokerCatalog -RemotePCDesktopGroupUid 42 | Remove-BrokerCatalog -RemotePCDesktopGroup 42
```

Remove all the Remote PC catalogs that are associated with desktop group 42. Note that this only breaks the Remote PC relationships and does not delete the desktop groups.

# Remove-BrokerCatalogMetadata

Jan 04, 2017

Deletes Catalog Metadata from the Catalog objects

## Syntax

```
Remove-BrokerCatalogMetadata [-InputObject] <Catalog[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerCatalogMetadata cmdlet deletes Metadata from the Catalog objects.

## Related topics

## Parameters

**-InputObject**<Catalog[]>

Specifies the Catalog object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerCatalog You can pipe the Catalog to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerCatalogMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the Catalog whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerCatalog | Remove-BrokerCatalogMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Catalog in the site

# Remove-BrokerConfigurationSlot

Jan 04, 2017

Removes a configuration slot.

## Syntax

```
Remove-BrokerConfigurationSlot [-InputObject] <ConfigurationSlot[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerConfigurationSlot [-Name] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Removes a configuration slot from the site. All machine configurations associated with this slot are also removed.

## Related topics

[New-BrokerConfigurationSlot](#)

[Get-BrokerConfigurationSlot](#)

## Parameters

**-InputObject**<ConfigurationSlot[]>

Configuration slot to remove.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Name of configuration slot to remove.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop

Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Configuration slot to remove Configuration slots may be specified through pipeline input.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

Remove-BrokerConfigurationSlot -Name "User Profile Manager"  
 Remove the configuration slot named "User Profile Manager".



# Remove-BrokerConfigurationSlotMetadata

Jan 04, 2017

Deletes ConfigurationSlot Metadata from the ConfigurationSlot objects

## Syntax

```
Remove-BrokerConfigurationSlotMetadata [-InputObject] <ConfigurationSlot[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerConfigurationSlotMetadata cmdlet deletes Metadata from the ConfigurationSlot objects.

## Related topics

## Parameters

**-InputObject**<ConfigurationSlot[]>

Specifies the ConfigurationSlot object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerConfigurationSlot You can pipe the ConfigurationSlot to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerConfigurationSlotMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the ConfigurationSlot whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerConfigurationSlot | Remove-BrokerConfigurationSlotMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the ConfigurationSlot in the site

# Remove-BrokerConfiguredFTA

Jan 04, 2017

Deletes one or more configured file type associations.

## Syntax

```
Remove-BrokerConfiguredFTA [-InputObject] <ConfiguredFTA[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Deletes one or more file type associations configured for a published application. At least one configured file type association object must be specified.

## Related topics

[Get-BrokerConfiguredFTA](#)

[New-BrokerConfiguredFTA](#)

## Parameters

**-InputObject**<ConfiguredFTA[]>

Specifies the ConfiguredFTA objects to delete.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.ConfiguredFTA[] One or more ConfiguredFTA objects can be supplied as input.

### Return Values

None

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> $ftas = Get-BrokerConfiguredFTA -ExtensionName ".txt"
```

```
C:\PS> Remove-BrokerConfiguredFTA $ftas
```

Deletes all configured file type associations with an extension name of ".txt".

# Remove-BrokerControllerMetadata

Jan 04, 2017

Deletes Controller Metadata from the Controller objects

## Syntax

```
Remove-BrokerControllerMetadata [-InputObject] <Controller[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerControllerMetadata cmdlet deletes Metadata from the Controller objects.

## Related topics

## Parameters

**-InputObject**<Controller[]>

Specifies the Controller object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerController You can pipe the Controller to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerControllerMetadata -InputObject \$obj-Uid -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for the Controller whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerController | Remove-BrokerControllerMetadata -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for all the Controller in the site

# Remove-BrokerDelayedHostingPowerAction

Jan 04, 2017

Cancels one or more delayed power actions.

## Syntax

```
Remove-BrokerDelayedHostingPowerAction [-InputObject] <DelayedHostingPowerAction[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerDelayedHostingPowerAction [-MachineName] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Removes one or more delayed power actions that have not yet been queued for execution.

## Related topics

[Get-BrokerDelayedHostingPowerAction](#)

[New-BrokerDelayedHostingPowerAction](#)

## Parameters

**-InputObject**<DelayedHostingPowerAction[]>

The power action to be cancelled.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-MachineName**<String>

Cancels only actions for machines whose name (of the form domain\machine) matches the specified string.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.DelayedHostingPowerAction The power action to be cancelled.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerHostingPowerAction -MachineName 'XD_VDA1'
```

Cancels any pending delayed power actions for the machine called XD\_VDA1.



# Remove-BrokerDesktopGroup

Jan 04, 2017

Remove desktop groups from the system or remove them from a Remote PC catalog.

## Syntax

```
Remove-BrokerDesktopGroup [-InputObject] <DesktopGroup[]> [-Force] [-RemotePCCatalog <Catalog>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerDesktopGroup [-Name] <String> [-Force] [-RemotePCCatalog <Catalog>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet has 2 functions:

- o Remove desktop groups from the system.
- o Break Remote PC associations between desktop groups and a catalog.

The Remote PC relationships are used by Remote PC automation to determine which desktop groups a machine in a particular Remote PC catalog can be published to. The assignment policy rules belonging to those desktop groups also determines the set of users that are allowed to be assigned to machines from the catalog.

## Related topics

[Get-BrokerDesktopGroup](#)

[New-BrokerDesktopGroup](#)

[Set-BrokerDesktopGroup](#)

[Add-BrokerDesktopGroup](#)

[Rename-BrokerDesktopGroup](#)

[New-BrokerCatalog](#)

[Remove-BrokerCatalog](#)

## Parameters

**-InputObject** <DesktopGroup[]>

Specifies the desktop groups to remove.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the desktop group to remove.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Force**<SwitchParameter>

Remove desktop groups even if there are active sessions.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-RemotePCCatalog**<Catalog>

When this parameter is specified, Remote PC desktop groups are removed from the specified Remote PC catalog.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.DesktopGroup You can pipe desktop groups to Remove-BrokerDesktopGroup.

**Return Values**

None

**Notes**

If a desktop group contains desktops when it is removed, these desktops are also removed (but the underlying broker machine remains).

A desktop group that still has active sessions cannot be removed unless the -Force switch is used.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerDesktopGroup EMEA*
Remove all desktop groups with names starting with "EMEA".
```

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerDesktopGroup -Enabled $false | Remove-BrokerDesktopGroup -Force
Remove all desktops that are currently disabled even if there are active sessions.
```

----- **EXAMPLE 3** -----

```
C:\PS> Get-BrokerDesktopGroup -RemotePCCatalogUid 42 | Remove-BrokerDesktopGroup -RemotePCCatalog 42
Remove all the Remote PC desktop groups that are associated with catalog 42. Note that this only breaks the Remote PC relationships and does not delete the desktop groups.
```

# Remove-BrokerDesktopGroupMetadata

Jan 04, 2017

Deletes DesktopGroup Metadata from the DesktopGroup objects

## Syntax

```
Remove-BrokerDesktopGroupMetadata [-InputObject] <DesktopGroup[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerDesktopGroupMetadata cmdlet deletes Metadata from the DesktopGroup objects.

## Related topics

## Parameters

**-InputObject**<DesktopGroup[]>

Specifies the DesktopGroup object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerDesktopGroup You can pipe the DesktopGroup to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerDesktopGroupMetadata -InputObject \$obj-Uid -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for the DesktopGroup whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerDesktopGroup | Remove-BrokerDesktopGroupMetadata -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for all the DesktopGroup in the site

# Remove-BrokerEntitlementPolicyRule

Jan 04, 2017

Deletes a desktop rule from the site's entitlement policy.

## Syntax

```
Remove-BrokerEntitlementPolicyRule [-InputObject] <EntitlementPolicyRule[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerEntitlementPolicyRule [-Name] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerEntitlementPolicyRule cmdlet deletes a desktop rule from the site's entitlement policy.

A desktop rule in the entitlement policy defines the users who are allowed per-session access to a machine from the rule's associated desktop group to run a full desktop session.

Deleting a rule does not affect existing sessions launched using the rule, but users cannot reconnect to those sessions if they are subsequently disconnected.

## Related topics

[New-BrokerEntitlementPolicyRule](#)

[Get-BrokerEntitlementPolicyRule](#)

[Set-BrokerEntitlementPolicyRule](#)

[Rename-BrokerEntitlementPolicyRule](#)

## Parameters

**-InputObject**<EntitlementPolicyRule[]>

The desktop rule to be deleted from the entitlement policy.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The name of the desktop rule to be deleted from the entitlement policy.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.EntitlementPolicyRule The desktop rule to be deleted from the entitlement policy.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerEntitlementPolicyRule 'Temp Workers'

Deletes the desktop rule called Temp Workers from the entitlement policy. Existing desktop sessions launched using the rule are not affected, but users cannot reconnect to sessions if they are subsequently disconnected.

----- **EXAMPLE 2** -----

```
C:\PS> $dg = Get-BrokerDesktopGroup 'Customer Support'
```

```
C:\PS> Get-BrokerEntitlementPolicyRule -DesktopGroupUid $dg.Uid | Remove-BrokerEntitlementPolicyRule
```

Deletes all desktop rules from the entitlement policy applying to the Customer Support desktop group. This effectively removes all access to the desktops published from this group. Existing desktop sessions are not affected, but users cannot reconnect to sessions if they are subsequently disconnected.



# Remove-BrokerEntitlementPolicyRuleMetadata

Jan 04, 2017

Deletes EntitlementPolicyRule Metadata from the EntitlementPolicyRule objects

## Syntax

```
Remove-BrokerEntitlementPolicyRuleMetadata [-InputObject] <EntitlementPolicyRule[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerEntitlementPolicyRuleMetadata cmdlet deletes Metadata from the EntitlementPolicyRule objects.

## Related topics

## Parameters

**-InputObject**<EntitlementPolicyRule[]>

Specifies the EntitlementPolicyRule object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerEntitlementPolicyRule You can pipe the EntitlementPolicyRule to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerEntitlementPolicyRuleMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the EntitlementPolicyRule whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerEntitlementPolicyRule | Remove-BrokerEntitlementPolicyRuleMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the EntitlementPolicyRule in the site

# Remove-BrokerHostingPowerAction

Jan 04, 2017

Cancel one or more pending power actions.

## Syntax

```
Remove-BrokerHostingPowerAction [-InputObject] <HostingPowerAction[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-BrokerHostingPowerAction [-MachineName] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Causes one or more of the pending power actions in the queue to be marked as cancelled. The affected power actions are not sent to the hypervisor for processing, and take no further part in the queuing activity.

Power actions cannot be cancelled once they have started to be processed by the hypervisor.

## Related topics

[Get-BrokerHostingPowerAction](#)

[New-BrokerHostingPowerAction](#)

[Set-BrokerHostingPowerAction](#)

## Parameters

**-InputObject** <HostingPowerAction[]>

The power action to be cancelled.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-MachineName** <String>

Cancels only actions for machines whose name (of the form domain\machine) matches the specified string.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId** <Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically

create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.HostingPowerAction The power action to be cancelled.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerHostingPowerAction -MachineName 'XD\_VDA1'  
 Cancels any pending power actions for the machine called 'XD\_VDA1'.

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerHostingPowerAction -Filter { State -eq "Pending" -and RequestTime -gt "-00:05" } | Remove-BrokerHostingPowerAction  
 Cancels any pending power actions requested in the last five minutes.

# Remove-BrokerHostingPowerActionMetadata

Jan 04, 2017

Deletes HostingPowerAction Metadata from the HostingPowerAction objects

## Syntax

```
Remove-BrokerHostingPowerActionMetadata [-InputObject] <HostingPowerAction[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerHostingPowerActionMetadata cmdlet deletes Metadata from the HostingPowerAction objects.

## Related topics

## Parameters

**-InputObject**<HostingPowerAction[]>

Specifies the HostingPowerAction object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerHostingPowerAction You can pipe the HostingPowerAction to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerHostingPowerActionMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the HostingPowerAction whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerHostingPowerAction | Remove-BrokerHostingPowerActionMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the HostingPowerAction in the site

# Remove-BrokerHypervisorAlertMetadata

Jan 04, 2017

Deletes HypervisorAlert Metadata from the HypervisorAlert objects

## Syntax

```
Remove-BrokerHypervisorAlertMetadata [-InputObject] <HypervisorAlert[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerHypervisorAlertMetadata cmdlet deletes Metadata from the HypervisorAlert objects.

## Related topics

## Parameters

**-InputObject**<HypervisorAlert[]>

Specifies the HypervisorAlert object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerHypervisorAlert You can pipe the HypervisorAlert to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerHypervisorAlertMetadata -InputObject \$obj-Uid -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for the HypervisorAlert whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerHypervisorAlert | Remove-BrokerHypervisorAlertMetadata -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for all the HypervisorAlert in the site



# Remove-BrokerHypervisorConnection

Jan 04, 2017

Removes a hypervisor connection from the system.

## Syntax

```
Remove-BrokerHypervisorConnection [-InputObject] <HypervisorConnection[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerHypervisorConnection [-Name] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Remove-BrokerHypervisorConnection removes a hypervisor connection from the system. A hypervisor connection cannot be removed if it's being used by a machine.

## Related topics

[Get-BrokerHypervisorConnection](#)

[Set-BrokerHypervisorConnection](#)

[New-BrokerHypervisorConnection](#)

## Parameters

**-InputObject**<HypervisorConnection[]>

Specifies the hypervisor connection object to remove.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the hypervisor connection object to remove.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.HypervisorConnection You can pipe the hypervisor connection to be removed to Remove-BrokerHypervisorConnection.

### Return Values

None

### Examples

#### ----- EXAMPLE 1 -----

```
c:\PS> Remove-BrokerHypervisorConnection -Name "Xen Server Connection"
```

This command removes a hypervisor connection by name.

#### ----- EXAMPLE 2 -----

```
c:\PS> $hvConn = Get-BrokerHypervisorConnection -PreferredController "controllerName" -Name "Xen Server Connection"
```

```
c:\PS> Remove-BrokerHypervisorConnection -InputObject $hvConn
```

Gets a hypervisor connection by preferred controller and removes it.

# Remove-BrokerHypervisorConnectionMetadata

Jan 04, 2017

Deletes HypervisorConnection Metadata from the HypervisorConnection objects

## Syntax

```
Remove-BrokerHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Name <String> [-LoggingId <Guid>]  
[-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerHypervisorConnectionMetadata cmdlet deletes Metadata from the HypervisorConnection objects.

## Related topics

## Parameters

**-InputObject**<HypervisorConnection[]>

Specifies the HypervisorConnection object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerHypervisorConnection You can pipe the HypervisorConnection to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerHypervisorConnectionMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the HypervisorConnection whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerHypervisorConnection | Remove-BrokerHypervisorConnectionMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the HypervisorConnection in the site

# Remove-BrokerIcon

Jan 04, 2017

Remove an icon.

## Syntax

```
Remove-BrokerIcon [-InputObject] <Icon[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Removes an icon from the database.

## Related topics

[Get-BrokerIcon](#)

[New-BrokerIcon](#)

[Set-BrokerIconMetadata](#)

## Parameters

**-InputObject**<Icon[]>

Specifies the icon to remove.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.Icon The icon to be removed can be piped into the cmdlet.

### Return Values

None

### Notes

Note that if the icon is currently in use, for example, as a desktop icon, it cannot be removed until the association is cleared.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerIcon 3  
Removes the icon with Uid 3.
```

# Remove-BrokerIconMetadata

Jan 04, 2017

Deletes Icon Metadata from the Icon objects

## Syntax

```
Remove-BrokerIconMetadata [-InputObject] <Icon[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

The Remove-BrokerIconMetadata cmdlet deletes Metadata from the Icon objects.

## Related topics

## Parameters

### **-InputObject**<Icon[]>

Specifies the Icon object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

### **-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerIcon You can pipe the Icon to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerIconMetadata -InputObject \$obj-Uid -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for the Icon whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerIcon | Remove-BrokerIconMetadata -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for all the Icon in the site



# Remove-BrokerImportedFTA

Jan 04, 2017

Deletes one or more imported file type associations.

## Syntax

```
Remove-BrokerImportedFTA -DesktopGroupUids <Int32[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Deletes all of the imported file type associations belonging to one or more desktop groups. At least one desktop group must be specified.

Imported file type associations are grouped together based on the desktop group of the machine from which they were imported. All file types for a desktop group are deleted. There is no mechanism for deleting a subset imported file type associations for a specific desktop group.

Imported file type associations are different from configured file type associations. Imported file type associations are lists of known file type associations for a given desktop group. Configured file type associations are those that are actually associated with published applications for the purposes of content redirection.

## Related topics

[Get-BrokerImportedFTA](#)

[Update-BrokerImportedFTA](#)

## Parameters

**-DesktopGroupUids**<Int32[]>

Deletes the imported file type associations belonging to specified desktop groups.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Int32[] An array of Uids for the desktop groups can be supplied as input. The desktop groups must be of the Private or Shared desktop kind.

**Return Values**

None

**Notes**

If an imported file type association is used to create a new configured file type association and the imported file type association is subsequently deleted, the configured file type association is not affected.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> $dg = Get-BrokerDesktopGroup -Name "Sales VMs"
C:\PS> Remove-BrokerImportedFTA -DesktopGroupUids $dg.Uid
Deletes all imported file type associations belonging to the "Sales VMs" desktop group.
```

# Remove-BrokerLease

Jan 04, 2017

Remove the specified lease in the Database.

## Syntax

```
Remove-BrokerLease [-InputObject] <Lease[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerLease [-Key] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Marks the specified lease for deletion. Note that the lease is eventually fully deleted when enough time has been allowed for the deletion to propagate to all controller machines in the site, but is immediately removed from lease search results.

## Related topics

[Update-BrokerLocalLeaseCache](#)

[Remove-BrokerCache](#)

## Parameters

**-InputObject**<Lease[]>

Specifies the lease to remove.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Key**<String>

Specifies the lease key of the lease to remove. A pattern can be specified.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop

Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.Lease The lease to be removed can be piped into the cmdlet.

**Return Values**

None

This cmdlet does not return any output.

**Notes**

The lease is marked for deletion after this cmdlet is run. Note that the lease is eventually fully deleted when enough time has been allowed for the deletion to propagate to all controller machines in the site, but is immediately removed from lease search results.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> $lease = Get-BrokerLease -Uid 1
C:\PS> Remove-BrokerLease $lease
Marks the specified lease for deletion.
```

# Remove-BrokerLeaseMetadata

Jan 04, 2017

Deletes Lease Metadata from the Lease objects

## Syntax

```
Remove-BrokerLeaseMetadata [-InputObject] <Lease[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

The Remove-BrokerLeaseMetadata cmdlet deletes Metadata from the Lease objects.

## Related topics

## Parameters

**-InputObject**<Lease[]>

Specifies the Lease object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerLease You can pipe the Lease to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerLeaseMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the Lease whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerLease | Remove-BrokerLeaseMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Lease in the site

# Remove-BrokerMachine

Jan 04, 2017

Removes one or more machines from its desktop group or catalog.

## Syntax

```
Remove-BrokerMachine [-InputObject] <Machine[]> [-Force] [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerMachine [-MachineName] <String> [-Force] [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerMachine cmdlet removes one or more machines from their desktop group or catalog. There are three forms:

- o Use the -InputObject parameter to remove a single machine instance or array of instances from their desktop group or catalog.
- o Use the -MachineName parameter to remove the single named machine from its group or catalog.
- o Use pipelining to pipe machine instances to the command.

To remove machines from their desktop group use the -DesktopGroup parameter; the specified group must be the one that contains the machines. If more than one machine is being removed from its group they must all be members of the same group.

If the -DesktopGroup parameter is not used then the machines are removed from their catalog. Removing a machine from its catalog deletes the record of the machine from the Citrix Broker Service.

Machines cannot be removed from their catalog while they are members of a desktop group.

## Related topics

[Add-BrokerMachine](#)

[Get-BrokerMachine](#)

## Parameters

**-InputObject** <Machine[]>

An array of machines to be removed from their desktop group or catalog.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-MachineName**<String>

The name of the single machine to remove (must match the MachineName property of the machine).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Force**<SwitchParameter>

Forces removal of machine from a desktop group even if it is still in use (that is, there are user sessions running on the machine). Forcing removal of a machine does not disconnect or logoff the user sessions.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroup**<DesktopGroup>

The desktop group from which the machines are to be removed, specified by name, UID, or instance.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |



|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.Machine You can pipe in the machines to be removed.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerMachine -InputObject $machine -DesktopGroup $desktopGroup
C:\PS> Remove-BrokerMachine -InputObject $machine -DesktopGroup 2
C:\PS> Remove-BrokerMachine $machine -DesktopGroup "MyDesktopGroup"
```

These all remove a single machine from a desktop group, identifying the group by instance, UID, or name.

----- **EXAMPLE 2** -----

```
C:\PS> Remove-BrokerMachine -MachineName "DOMAIN\MyMachine" -DesktopGroup 2
C:\PS> Remove-BrokerMachine DOMAIN\MyMachine -DesktopGroup "MyDesktopGroup"
C:\PS> Remove-BrokerMachine DOMAIN\MyMachine -DesktopGroup $desktopGroup
```

These remove the machine called "DOMAIN\MyMachine" from its desktop group.

----- **EXAMPLE 3** -----

```
C:\PS> Remove-BrokerMachine -MachineName DOMAIN\MyMachine
C:\PS> Remove-BrokerMachine "DOMAIN\MyMachine"
C:\PS> Remove-BrokerMachine $machine
```

These all remove a machine from its catalog.

----- **EXAMPLE 4** -----

```
C:\PS> Get-BrokerMachine -Uid 3 | Remove-BrokerMachine -DesktopGroup $dg
C:\PS> Get-BrokerMachine -CatalogUid 4 | Remove-BrokerMachine
```

These find specific machines and remove them from their desktop group or catalog.

# Remove-BrokerMachineCommand

Jan 04, 2017

Cancel a pending command queued for delivery to a desktop.

## Syntax

```
Remove-BrokerMachineCommand [-InputObject] <MachineCommand[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Sets the state of a pending command queued for delivery to a desktop to Canceled. The command is not removed from the system.

## Related topics

[Get-BrokerMachineCommand](#)

[New-BrokerMachineCommand](#)

## Parameters

**-InputObject**<MachineCommand[]>

Commands to cancel.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.MachineCommand Commands to cancel.

### Return Values

None

### Examples

----- **EXAMPLE 1** -----

Get-BrokerMachineCommand | Remove-BrokerMachineCommand  
Cancel all pending commands.

----- **EXAMPLE 2** -----

Get-BrokerMachineCommand -Category "UPM" | Remove-BrokerMachineCommand  
Cancel all pending commands that have the category "UPM".

# Remove-BrokerMachineCommandMetadata

Jan 04, 2017

Deletes MachineCommand Metadata from the MachineCommand objects

## Syntax

```
Remove-BrokerMachineCommandMetadata [-InputObject] <MachineCommand[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerMachineCommandMetadata cmdlet deletes Metadata from the MachineCommand objects.

## Related topics

## Parameters

**-InputObject**<MachineCommand[]>

Specifies the MachineCommand object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerMachineCommand You can pipe the MachineCommand to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerMachineCommandMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the MachineCommand whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerMachineCommand | Remove-BrokerMachineCommandMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the MachineCommand in the site

# Remove-BrokerMachineConfiguration

Jan 04, 2017

Deletes a machine configuration from the site or removes the association from a desktop group.

## Syntax

```
Remove-BrokerMachineConfiguration [-InputObject] <MachineConfiguration[]> [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerMachineConfiguration [-Name] <String> [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-MachineConfiguration cmdlet deletes machine configurations from the site. A machine configuration cannot be removed while it is applied to a desktop group.

When a desktop group is provided, the Remove-MachineConfiguration cmdlet removes the association between machine configuration and the desktop group. In this case, the machine configuration is not removed from the site.

## Related topics

[New-BrokerMachineConfiguration](#)

[Get-BrokerMachineConfiguration](#)

[Set-BrokerMachineConfiguration](#)

[Rename-BrokerMachineConfiguration](#)

[Add-BrokerMachineConfiguration](#)

## Parameters

**-InputObject**<MachineConfiguration[]>

Machine configuration to remove.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          | None           |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Name of machine configuration for which the remove operation applies.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          | None                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-DesktopGroup**<DesktopGroup>

The desktop group from which this machine configuration is to be removed.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          | None           |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.MachineConfiguration Machine configuration to remove

## Return Values

None

## Notes

A machine configuration can only be removed if it is not currently applied to a desktop group.

## Examples

### ----- EXAMPLE 1 -----

```
Remove-BrokerMachineConfiguration -Name UPM\Finance
```

Removes the machine configuration named "UPM\Finance".

### ----- EXAMPLE 2 -----

```
Remove-BrokerMachineConfiguration -Name Receiver\HumanResources -DesktopGroup SharedWorkers
```

Removes the association of the machine configuration named "Receiver\HumanResources" from the "SharedWorkers" desktop group.



# Remove-BrokerMachineConfigurationMetadata

Jan 04, 2017

Deletes MachineConfiguration Metadata from the MachineConfiguration objects

## Syntax

```
Remove-BrokerMachineConfigurationMetadata [-InputObject] <MachineConfiguration[]> -Name <String> [-LoggingId <Guid>]  
[-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerMachineConfigurationMetadata cmdlet deletes Metadata from the MachineConfiguration objects.

## Related topics

## Parameters

**-InputObject**<MachineConfiguration[]>

Specifies the MachineConfiguration object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerMachineConfiguration You can pipe the MachineConfiguration to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerMachineConfigurationMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the MachineConfiguration whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerMachineConfiguration | Remove-BrokerMachineConfigurationMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the MachineConfiguration in the site

# Remove-BrokerMachineMetadata

Jan 04, 2017

Deletes Machine Metadata from the Machine objects

## Syntax

```
Remove-BrokerMachineMetadata [-InputObject] <Machine[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

The Remove-BrokerMachineMetadata cmdlet deletes Metadata from the Machine objects.

## Related topics

## Parameters

**-InputObject**<Machine[]>

Specifies the Machine object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerMachine You can pipe the Machine to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerMachineMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the Machine whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerMachine | Remove-BrokerMachineMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Machine in the site

# Remove-BrokerPowerTimeScheme

Jan 04, 2017

Deletes an existing power time scheme.

## Syntax

```
Remove-BrokerPowerTimeScheme [-InputObject] <PowerTimeScheme[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerPowerTimeScheme [-Name] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerPowerTimeScheme cmdlet deletes a power time scheme from the system, and leaves the days that the time scheme used to cover for the associated desktop group as defaulting to all hours off-peak and all hours with pool size of -1.

Each power time scheme is associated with a particular desktop group, and covers one or more days of the week, defining which hours of those days are considered peak times and which are off-peak times. In addition, the time scheme defines a pool size value for each hour of the day for the days of the week covered by the time scheme. No one desktop group can be associated with two or more time schemes that cover the same day of the week.

For more information about the power policy mechanism and pool size management, see 'help about\_Broker\_PowerManagement'.

## Related topics

[Get-BrokerPowerTimeScheme](#)

[Set-BrokerPowerTimeScheme](#)

[New-BrokerPowerTimeScheme](#)

[Rename-BrokerPowerTimeScheme](#)

## Parameters

**-InputObject**<PowerTimeScheme[]>

The power time scheme to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The name of the power time scheme to be deleted.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High

Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.PowerTimeScheme The power time scheme to be deleted.

Return Values

None

Examples

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerPowerTimeScheme -Name 'Development Weekdays'  
 Deletes the power time scheme named 'Development Weekdays'.

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerPowerTimeScheme -DesktopGroupUid (Get-BrokerDesktopGroup -name 'Finance desk1').Uid | Remove-BrokerPowerTimeScheme  
 Deletes all power time schemes for the desktop group named 'Finance desk1'.

# Remove-BrokerPowerTimeSchemeMetadata

Jan 04, 2017

Deletes PowerTimeScheme Metadata from the PowerTimeScheme objects

## Syntax

```
Remove-BrokerPowerTimeSchemeMetadata [-InputObject] <PowerTimeScheme[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerPowerTimeSchemeMetadata cmdlet deletes Metadata from the PowerTimeScheme objects.

## Related topics

## Parameters

**-InputObject**<PowerTimeScheme[]>

Specifies the PowerTimeScheme object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerPowerTimeScheme You can pipe the PowerTimeScheme to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerPowerTimeSchemeMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the PowerTimeScheme whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerPowerTimeScheme | Remove-BrokerPowerTimeSchemeMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the PowerTimeScheme in the site



# Remove-BrokerRebootCycleMetadata

Jan 04, 2017

Deletes RebootCycle Metadata from the RebootCycle objects

## Syntax

```
Remove-BrokerRebootCycleMetadata [-InputObject] <RebootCycle[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerRebootCycleMetadata cmdlet deletes Metadata from the RebootCycle objects.

## Related topics

## Parameters

**-InputObject**<RebootCycle[]>

Specifies the RebootCycle object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerRebootCycle You can pipe the RebootCycle to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerRebootCycleMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the RebootCycle whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerRebootCycle | Remove-BrokerRebootCycleMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the RebootCycle in the site

# Remove-BrokerRebootSchedule

Jan 04, 2017

Removes the reboot schedule.

## Syntax

```
Remove-BrokerRebootSchedule [-InputObject] <RebootSchedule[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-BrokerRebootSchedule [-DesktopGroupName] <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

The Remove-BrokerRebootSchedule cmdlet is used to delete an existing reboot schedule.

## Related topics

[Get-BrokerRebootSchedule](#)

[Set-BrokerRebootSchedule](#)

[New-BrokerRebootSchedule](#)

[Stop-BrokerRebootCycle](#)

## Parameters

**-InputObject**<RebootSchedule[]>

The reboot schedule to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-DesktopGroupName**<String>

The name of the desktop group whose reboot schedule is to be removed.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.RebootSchedule Reboot schedules may be specified through pipeline input.

### Return Values

None

### Examples

#### ----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerRebootSchedule | Remove-BrokerRebootSchedule  
Deletes every reboot schedule.
```

#### ----- EXAMPLE 2 -----

```
C:\PS> Remove-BrokerRebootSchedule 12  
Deletes the reboot schedule for the desktop group having Uid 12.
```

#### ----- EXAMPLE 3 -----

```
C:\PS> Remove-BrokerRebootSchedule -DesktopGroupName Accounting  
Deletes the reboot schedule for the desktop group named Accounting.
```



# Remove-BrokerRemotePCAccount

Jan 04, 2017

Delete RemotePCAccounts from the system.

## Syntax

```
Remove-BrokerRemotePCAccount [-InputObject] <RemotePCAccount[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Delete RemotePCAccounts from the site.

## Related topics

[Get-BrokerRemotePCAccount](#)

[New-BrokerRemotePCAccount](#)

[Set-BrokerRemotePCAccount](#)

## Parameters

**-InputObject**<RemotePCAccount[]>

Specifies the RemotePCAccounts to delete.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.RemotePCAccount You can pipe the RemotePCAccounts to be deleted into this cmdlet.

### Return Values

None

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerRemotePCAccount 42  
Delete RemotePCAccount 42.
```

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerRemotePCAccount -OU 'any' | Remove-BrokerRemotePCAccount  
Delete the 'any' OU RemotePCAccount.
```

----- **EXAMPLE 3** -----

```
C:\PS> Get-BrokerRemotePCAccount | Remove-BrokerRemotePCAccount  
Delete all RemotePCAccounts.
```

# Remove-BrokerScope

Jan 04, 2017

Remove the specified catalog/desktop group from the given scope(s).

## Syntax

```
Remove-BrokerScope [-InputObject] <Scope[]> [-Catalog <Catalog>] [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerScope cmdlet is used to remove a catalog/desktop group object from the given scope(s).

To remove a catalog/desktop group from a scope you need permission to change the scopes of the catalog/desktop group.

If the catalog/desktop group is not in a specified scope, that scope will be silently ignored.

## Related topics

### Parameters

**-InputObject**<Scope[]>

Specifies the scopes to remove the object from.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-Catalog**<Catalog>

Specifies the catalog object to be removed.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-DesktopGroup**<DesktopGroup>

Specifies the desktop group object to be removed.



|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.Scope You can pipe scopes to Remove-BrokerScope.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerScope -Scope Chalfont,Cambourne -DesktopGroup "Win7 Desktops"  
Removes the "Win7 Desktops" desktop group from both the Chalfont and Cambourne scopes.

----- **EXAMPLE 2** -----

C:\PS> 'Chalfont','Cambourne' | Remove-BrokerScope -DesktopGroup 'Win7 Desktops'  
Removes the "Win7 Desktops" desktop group from both the Chalfont and Cambourne scopes.

# Remove-BrokerSessionLinger

Jan 04, 2017

Removes a session linger setting.

## Syntax

```
Remove-BrokerSessionLinger [-InputObject] <SessionLinger[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-BrokerSessionLinger [-DesktopGroupName] <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

The Remove-BrokerSessionLinger cmdlet is used to delete an existing session linger setting.

## Related topics

[New-BrokerSessionLinger](#)

[Get-BrokerSessionLinger](#)

[Set-BrokerSessionLinger](#)

## Parameters

**-InputObject** <SessionLinger[]>

The session linger setting to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-DesktopGroupName** <String>

The name of the desktop group whose session linger setting is to be removed.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.SessionLinger Session linger settings may be specified through pipeline input.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerSessionLinger | Remove-BrokerSessionLinger  
Deletes every session linger setting for all desktop groups.
```

# Remove-BrokerSessionMetadata

Jan 04, 2017

Deletes Session Metadata from the Session objects

## Syntax

```
Remove-BrokerSessionMetadata [-InputObject] <Session[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

The Remove-BrokerSessionMetadata cmdlet deletes Metadata from the Session objects.

## Related topics

## Parameters

**-InputObject**<Session[]>

Specifies the Session object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerSession You can pipe the Session to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerSessionMetadata -InputObject $obj-Uid -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for the Session whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerSession | Remove-BrokerSessionMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Session in the site

# Remove-BrokerSessionPreLaunch

Jan 04, 2017

Removes a session pre-launch setting.

## Syntax

```
Remove-BrokerSessionPreLaunch [-InputObject] <SessionPreLaunch[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerSessionPreLaunch [-DesktopGroupName] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerSessionPreLaunch cmdlet is used to delete an existing session pre-launch setting.

## Related topics

[New-BrokerSessionPreLaunch](#)

[Get-BrokerSessionPreLaunch](#)

[Set-BrokerSessionPreLaunch](#)

## Parameters

**-InputObject** <SessionPreLaunch[]>

The session pre-launch setting to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-DesktopGroupName** <String>

The name of the desktop group whose session pre-launch setting is to be removed.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.SessionPreLaunch Session pre-launch settings may be specified through pipeline input.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerSessionPreLaunch | Remove-BrokerSessionPreLaunch  
Deletes every session pre-launch setting for all desktop groups.
```

# Remove-BrokerSiteMetadata

Jan 04, 2017

Deletes Site Metadata from the Site objects

## Syntax

```
Remove-BrokerSiteMetadata -Name <String> [[-InputObject] <Site[]>] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

The Remove-BrokerSiteMetadata cmdlet deletes Metadata from the Site objects.

## Related topics

## Parameters

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InputObject**<Site[]>

Specifies the Site object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerSite You can pipe the Site to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerSiteMetadata -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for the Site

# Remove-BrokerTag

Jan 04, 2017

Removes a tag from the system or clears a tag to object association.

## Syntax

```
Remove-BrokerTag [-InputObject] <Tag[]> [-Desktop <Desktop>] [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerTag [-Name] <String> [-Desktop <Desktop>] [-DesktopGroup <DesktopGroup>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Clears tag to object associations or delete tags from the system altogether.

To clear an association, supply one of the Application, DesktopGroup or PrivateDesktop parameters.

To delete the tag, and any associations between the tag and other objects in the database, specify the tag without any associated object parameters.

## Related topics

[Add-BrokerTag](#)

[Get-BrokerTag](#)

[New-BrokerTag](#)

[Rename-BrokerTag](#)

## Parameters

**-InputObject**<Tag[]>

Specifies one or more tag objects.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies a tag by name.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Desktop**<Desktop>

Clears the association between the given tag and Desktop.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-DesktopGroup**<DesktopGroup>

Clears the association between the given tag and desktop group.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.Tag Tags may be specified through pipeline input.

### Return Values

None

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-BrokerTag $tag -Desktop $desktop
```

Clears the association between a tag and a desktop. Note that the tag itself continues to exist as an object in the database.

----- **EXAMPLE 2** -----

```
C:\PS> Remove-BrokerTag $tag
```

Deletes the tag object from the database and clears any associations that may exist between that tag and other objects.

# Remove-BrokerTagMetadata

Jan 04, 2017

Deletes Tag Metadata from the Tag objects

## Syntax

```
Remove-BrokerTagMetadata [-InputObject] <Tag[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

The Remove-BrokerTagMetadata cmdlet deletes Metadata from the Tag objects.

## Related topics

## Parameters

**-InputObject**<Tag[]>

Specifies the Tag object's instance whose Metadata is to be deleted.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata to be deleted

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerTag You can pipe the Tag to delete the metadata.

Return Values

None

Examples

----- **EXAMPLE 1** -----

C:\PS> Remove-BrokerTagMetadata -InputObject \$obj-Uid -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for the Tag whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerTag | Remove-BrokerTagMetadata -Name "MyMetadataName"

This command deletes the Metadata "MyMetadataName" key-value pair for all the Tag in the site

# Remove-BrokerUser

Jan 04, 2017

Remove broker user objects from another broker object

## Syntax

```
Remove-BrokerUser [-InputObject] <User[]> [-Application <Application>] [-SessionLinger <SessionLinger>] [-SessionPreLaunch <SessionPreLaunch>] [-Machine <Machine>] [-PrivateDesktop <PrivateDesktop>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerUser [-Name] <String> [-Application <Application>] [-SessionLinger <SessionLinger>] [-SessionPreLaunch <SessionPreLaunch>] [-Machine <Machine>] [-PrivateDesktop <PrivateDesktop>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-BrokerUser cmdlet removes broker user objects from another specified object, such as a broker private desktop, to which the user had previously been added.

## Related topics

[Add-BrokerUser](#)

[Get-BrokerUser](#)

## Parameters

**-InputObject** <User[]>

Specifies the user objects to remove.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name** <String>

Specifies the user objects to remove, based on their Name property.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Application**<Application>

The application from which to remove the user

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-SessionLinger**<SessionLinger>

The desktop group session linger setting from which to remove the user.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-SessionPreLaunch**<SessionPreLaunch>

The desktop group session pre-launch setting from which to remove the user.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-Machine**<Machine>

The machine from which to remove the user

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-PrivateDesktop**<PrivateDesktop>



The desktop from which to remove the user

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

#### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

#### Input Type

Citrix.Broker.Admin.SDK.User You can pipe the users to be removed to Remove-BrokerUser.

#### Return Values

None

#### Notes

Specify one of the -Machine or -PrivateDesktop parameters only.

#### Examples

----- **EXAMPLE 1** -----

```
Remove-BrokerUser "DOMAIN\UserName" -PrivateDesktop "DOMAIN\MachineName"
```

Remove the assignment of the specified private desktop to the specified user.

# Rename-BrokerAccessPolicyRule

Jan 04, 2017

Renames a rule in the site's access policy.

## Syntax

```
Rename-BrokerAccessPolicyRule [-InputObject] <AccessPolicyRule[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-BrokerAccessPolicyRule [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Rename-BrokerAccessPolicyRule cmdlet renames a rule in the site's access policy. The Name property of the rule is changed.

An access policy rule defines a set of connection filters and access control rights relating to a desktop group. These allow fine-grained control of what access is granted to a desktop group based on details of, for example, a user's endpoint device, its address, and the user's identity.

## Related topics

[New-BrokerAccessPolicyRule](#)

[Get-BrokerAccessPolicyRule](#)

[Set-BrokerAccessPolicyRule](#)

[Remove-BrokerAccessPolicyRule](#)

## Parameters

**-InputObject**<AccessPolicyRule[]>

The access policy rule to be renamed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The existing name of the access policy rule to be renamed.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-NewName**<String>

The new name for the access policy rule being renamed. The new name must not match that of any other existing rules in the policy.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host

name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.AccessPolicyRule The access policy rule to be renamed.

### Return Values

None or Citrix.Broker.Admin.SDK.AccessPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AccessPolicyRule object.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Rename-BrokerAccessPolicyRule 'Sales' -NewName 'TeleSales'
```

Renames the access policy rule called Sales to TeleSales. The new name of the rule must be unique in the access policy.

# Rename-BrokerAdminFolder

Jan 04, 2017

Renames a folder

## Syntax

```
Rename-BrokerAdminFolder [-InputObject] <AdminFolder[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-BrokerAdminFolder [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Rename-BrokerAdminFolder cmdlet renames a folder for organising objects for administration purposes (for example, Applications).

The following special characters are not allowed in the new FolderName: \ / ; : # . \* ? = < > | [ ] ( ) " ' `

## Related topics

[Get-BrokerAdminFolder](#)

[New-BrokerAdminFolder](#)

## Parameters

**-InputObject**<AdminFolder[]>

The folder(s) to be renamed

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

A pattern matching the names of folders to be renamed

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-NewName**<String>

The name the new folder(s) should have.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|               |                                                                                        |
|---------------|----------------------------------------------------------------------------------------|
| Required?     | false                                                                                  |
| Default Value | localhost. Once a value is provided by any cmdlet, this value will become the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Input Type

Depends on parameter Parameters can be piped by property name.

## Return Values

None or Citrix.Broker.Admin.SDK.AdminFolder

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AdminFolder object.

## Examples

----- **EXAMPLE 1** -----

Rename-BrokerAdminFolder F1\XXX\ YYY

Renames the folder called XXX within the folder F1\ to YYY



# Rename-BrokerAppAssignmentPolicyRule

Jan 04, 2017

Renames an application rule in the site's assignment policy.

## Syntax

```
Rename-BrokerAppAssignmentPolicyRule [-InputObject] <AppAssignmentPolicyRule[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-BrokerAppAssignmentPolicyRule [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Rename-BrokerAppAssignmentPolicyRule cmdlet renames an application rule in the site's assignment policy. The Name property of the rule is changed.

An application rule in the assignment policy defines the users who are entitled to a self-service persistent machine assignment from the rule's desktop group; once assigned the machine can run one or more applications published from the group.

## Related topics

[New-BrokerAppAssignmentPolicyRule](#)

[Get-BrokerAppAssignmentPolicyRule](#)

[Set-BrokerAppAssignmentPolicyRule](#)

[Remove-BrokerAppAssignmentPolicyRule](#)

## Parameters

**-InputObject** <AppAssignmentPolicyRule[]>

The application rule in the assignment policy to be renamed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name** <String>

The existing name of the application rule in the assignment policy to be renamed.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-NewName**<String>

The new name of the application rule in the assignment policy being renamed. The new name must not match that of any other existing rule in the policy (irrespective of whether it is a desktop or application rule).

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host

name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.AppAssignmentPolicyRule The application rule in the assignment policy being renamed.

### Return Values

None or Citrix.Broker.Admin.SDK.AppAssignmentPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AppAssignmentPolicyRule object.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Rename-BrokerAppAssignmentPolicyRule 'Offshore' -NewName 'Remote Workers'
```

Renames the application rule in the assignment policy called Offshore to Remote Workers. The new name of the rule must be unique in the assignment policy.

# Rename-BrokerAppEntitlementPolicyRule

Jan 04, 2017

Renames an application rule in the site's entitlement policy.

## Syntax

```
Rename-BrokerAppEntitlementPolicyRule [-InputObject] <AppEntitlementPolicyRule[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-BrokerAppEntitlementPolicyRule [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Rename-BrokerAppEntitlementPolicyRule cmdlet renames an application rule in the site's entitlement policy. The Name property of the rule is changed.

An application rule in the entitlement policy defines the users who are allowed per-session access to a machine to run one or more applications published from the rule's desktop group.

## Related topics

[New-BrokerAppEntitlementPolicyRule](#)

[Get-BrokerAppEntitlementPolicyRule](#)

[Set-BrokerAppEntitlementPolicyRule](#)

[Remove-BrokerAppEntitlementPolicyRule](#)

## Parameters

**-InputObject** <AppEntitlementPolicyRule[]>

The application rule in the entitlement policy to be renamed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name** <String>

The existing name of the application rule in the entitlement policy to be renamed.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-NewName**<String>

The new name of the application rule in the entitlement policy being renamed. The new name must not match that of any other existing rule in the policy (irrespective of whether it is a desktop or application rule).

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule The application rule in the entitlement policy being renamed.

### Return Values

None or Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule object.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Rename-BrokerAppEntitlementPolicyRule 'Prod Dev' -NewName 'Product Development'
```

Renames the application rule in the entitlement policy called Prod Dev to Product Development. The new name of the rule must be unique in the entitlement policy.

# Rename-BrokerApplication

Jan 04, 2017

Renames an application.

## Syntax

```
Rename-BrokerApplication [-InputObject] <Application[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-BrokerApplication [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Rename-BrokerApplication cmdlet changes the administrative name of an application. An application cannot have the same name as another application.

Renaming an application does not alter its published name. To change the name with which this application appears to end-users, set a new value for the PublishedName property using the Set-BrokerApplication cmdlet.

Renaming an application does not alter its BrowserName. If the BrowserName property also needs to be changed, use the Set-BrokerApplication cmdlet to modify it.

## Related topics

[New-BrokerApplication](#)

[Set-BrokerApplication](#)

[Get-BrokerApplication](#)

[Remove-BrokerApplication](#)

## Parameters

**-InputObject** <Application[]>

Specifies the application to rename.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-Name** <String>

Specifies the name of the application to rename.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          | null                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-NewName**<String>

Specifies the new name for the application.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host



name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.Application You can pipe applications to Rename-BrokerApplication.

### Return Values

None or Citrix.Broker.Admin.SDK.Application

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.Application object.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Rename-BrokerApplication -Name "Old Name" -NewName "New Name"
```

Renames the application with name "Old Name" to "New Name".

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerApplication -Uid 1 | Rename-BrokerApplication -NewName "New Name" -PassThru
```

Renames application with the Uid 1 to "New Name", showing the result.

# Rename-BrokerAssignmentPolicyRule

Jan 04, 2017

Renames a desktop rule in the site's assignment policy.

## Syntax

```
Rename-BrokerAssignmentPolicyRule [-InputObject] <AssignmentPolicyRule[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-BrokerAssignmentPolicyRule [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Rename-BrokerAssignmentPolicyRule cmdlet renames a desktop rule in the site's assignment policy. The Name property of the rule is changed.

A desktop rule in the assignment policy defines the users who are entitled to self-service persistent machine assignments from the rule's desktop group. A rule defines how many machines a user is allowed from the group for delivery of full desktop sessions.

## Related topics

[New-BrokerAssignmentPolicyRule](#)

[Get-BrokerAssignmentPolicyRule](#)

[Set-BrokerAssignmentPolicyRule](#)

[Remove-BrokerAssignmentPolicyRule](#)

## Parameters

**-InputObject** <AssignmentPolicyRule[]>

The desktop rule in the assignment policy to be renamed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name** <String>

The existing name of the desktop rule in the assignment policy to be renamed.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-NewName**<String>

The new name of the desktop rule in the assignment policy being renamed. The new name must not match that of any other existing rule in the policy (irrespective of whether it is a desktop or application rule).

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host

name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.AssignmentPolicyRule The desktop rule in the assignment policy being renamed.

### Return Values

None, or Citrix.Broker.Admin.SDK.AssignmentPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AssignmentPolicyRule object.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Rename-BrokerAssignmentPolicyRule 'Offshore' -NewName 'Remote Workers'
```

Renames the desktop rule in the assignment policy called Offshore to Remote Workers. The new name of the rule must be unique in the assignment policy.

# Rename-BrokerCatalog

Jan 04, 2017

Renames a catalog.

## Syntax

```
Rename-BrokerCatalog [-InputObject] <Catalog[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-BrokerCatalog [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Rename-BrokerCatalog cmdlet changes the name of a catalog. A catalog cannot have the same name as another catalog.

The following special characters are not allowed in a catalog name: \ / ; : # . \* ? = < > | [ ] ( ) ' ' `

## Related topics

[Get-BrokerCatalog](#)

[New-BrokerCatalog](#)

[Remove-BrokerCatalog](#)

[Set-BrokerCatalog](#)

## Parameters

**-InputObject**<Catalog[]>

Specifies the catalog to rename.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the catalog to rename.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-NewName**<String>

Specifies the new name of the catalog.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

Citrix.Broker.Admin.SDK.Catalog You can pipe catalogs to Rename-BrokerCatalog.

## Return Values

None or Citrix.Broker.Admin.SDK.Catalog

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.Catalog object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Rename-BrokerCatalog -Name "Old Name" -NewName "New Name"
Renames the catalog with the name "Old Name" to "New Name".
```

### ----- EXAMPLE 2 -----

```
C:\PS> c:\$catalog = Get-BrokerCatalog -Name "Old Name"
C:\PS> Rename-BrokerCatalog -InputObject $catalog -NewName "New Name"
Renames the catalog with the name "Old Name" to "New Name".
```

# Rename-BrokerDesktopGroup

Jan 04, 2017

Renames a desktop group.

## Syntax

```
Rename-BrokerDesktopGroup [-InputObject] <DesktopGroup[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-BrokerDesktopGroup [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Rename-BrokerDesktopGroup cmdlet changes the name of a desktop group. A desktop group cannot have the same name as another desktop group.

## Related topics

[Get-BrokerDesktopGroup](#)

[New-BrokerDesktopGroup](#)

[Set-BrokerDesktopGroup](#)

[Remove-BrokerDesktopGroup](#)

## Parameters

**-InputObject** <DesktopGroup[]>

Specifies the desktop group to rename.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          | null           |
| Accept Pipeline Input? | true (ByValue) |

**-Name** <String>

Specifies the name of the desktop group to rename.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value | null |



|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-NewName**<String>

Specifies the new name that the desktop group will have.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|           |       |

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

Citrix.Broker.Admin.SDK.DesktopGroup You can pipe desktop groups to Rename-BrokerDesktopGroup.

## Return Values

None or Citrix.Broker.Admin.SDK.DesktopGroup

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.DesktopGroup object.

## Notes

Renaming a desktop group does not alter its published name. If you need to change the name with which this desktop group appears to end-users, set a new value for the PublishedName property using the Set-BrokerDesktopGroup cmdlet.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Rename-BrokerDesktopGroup -Name "Old Name" -NewName "New Name"
Renames desktop group with the name "Old Name" to "New Name".
```

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerDesktopGroup -Uid 1 | Rename-BrokerDesktopGroup -NewName "New Name" -PassThru
Renames desktop group with the Uid 1 to "New Name", showing the result.
```

# Rename-BrokerEntitlementPolicyRule

Jan 04, 2017

Renames a desktop rule in the site's entitlement policy.

## Syntax

```
Rename-BrokerEntitlementPolicyRule [-InputObject] <EntitlementPolicyRule[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-BrokerEntitlementPolicyRule [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Rename-BrokerEntitlementPolicyRule cmdlet renames a desktop rule in the site's entitlement policy. The Name property of the rule is changed.

A desktop rule in the entitlement policy defines the users who are allowed per-session access to a machine from the rule's associated desktop group to run a full desktop session.

## Related topics

[New-BrokerEntitlementPolicyRule](#)

[Get-BrokerEntitlementPolicyRule](#)

[Set-BrokerEntitlementPolicyRule](#)

[Remove-BrokerEntitlementPolicyRule](#)

## Parameters

**-InputObject**<EntitlementPolicyRule[]>

The desktop rule in the entitlement policy to be renamed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The existing name of the desktop rule in the entitlement policy to be renamed.

|                        |      |
|------------------------|------|
| Required?              | true |
| Default Value          |      |
| Accept Pipeline Input? |      |

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-NewName**<String>

The new name of the desktop rule in the entitlement policy being renamed. The new name must not match that of any other existing rule in the policy (irrespective of whether it is a desktop or application rule).

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

Citrix.Broker.Admin.SDK.EntitlementPolicyRule The desktop rule in the entitlement policy being renamed.

## Return Values

None or Citrix.Broker.Admin.SDK.EntitlementPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.EntitlementPolicyRule object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Rename-BrokerEntitlementPolicyRule 'Prod Dev' -NewName 'Product Development'
```

Renames the desktop rule in the entitlement policy called Prod Dev to Product Development. The new name of the rule must be unique in the entitlement policy.

# Rename-BrokerMachineConfiguration

Jan 04, 2017

Renames a machine configuration.

## Syntax

```
Rename-BrokerMachineConfiguration [-InputObject] <MachineConfiguration[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-BrokerMachineConfiguration [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Rename-MachineConfiguration cmdlet changes the name of a machine configuration. A machine configuration cannot have the same name as another machine configuration associated with the same slot.

## Related topics

[New-BrokerMachineConfiguration](#)

[Get-BrokerMachineConfiguration](#)

[Set-BrokerMachineConfiguration](#)

[Remove-BrokerMachineConfiguration](#)

[Add-BrokerMachineConfiguration](#)

## Parameters

**-InputObject** <MachineConfiguration[]>

Machine configuration to rename.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          | None           |
| Accept Pipeline Input? | true (ByValue) |

**-Name** <String>

Current name of machine configuration.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value | None |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-NewName**<String>

New name for machine configuration. This may have the form "ConfigurationSlotName\MachineConfigurationName" or "MachineConfigurationName". If the "ConfigurationSlotName" is provided it must match the name of the configuration slot that the machine configuration is associated with.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          | None  |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

Citrix.Broker.Admin.SDK.MachineConfiguration Machine configuration to rename.

## Return Values

None or Citrix.Broker.Admin.SDK.MachineConfiguration

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.MachineConfiguration object.

## Notes

The configuration slot can not be changed. Thus the left term of the Name and NewName must match.

## Examples

### ----- EXAMPLE 1 -----

Rename-BrokerMachineConfiguration -Name "UPM\All Departments" -NewName "UPM\Finance Department"  
 Renames the machine configuration named "UPM\All Departments" to "UPM\Finance Department".

### ----- EXAMPLE 2 -----

Rename-BrokerMachineConfiguration -Name "UPM\All Departments" -NewName "Finance Department"  
 Renames the machine configuration named "UPM\All Departments" to "UPM\Finance Department".



# Rename-BrokerPowerTimeScheme

Jan 04, 2017

Changes the name of an existing power time scheme.

## Syntax

```
Rename-BrokerPowerTimeScheme [-InputObject] <PowerTimeScheme[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-BrokerPowerTimeScheme [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Rename-BrokerPowerTimeScheme cmdlet renames a particular power time scheme.

Each power time scheme is associated with a particular desktop group, and covers one or more days of the week, defining which hours of those days are considered peak times and which are off-peak times. In addition, the time scheme defines a pool size value for each hour of the day for the days of the week covered by the time scheme. No one desktop group can be associated with two or more time schemes that cover the same day of the week.

For more information about the power policy mechanism and pool size management, see 'help about\_Broker\_PowerManagement'.

## Related topics

[Get-BrokerPowerTimeScheme](#)

[Set-BrokerPowerTimeScheme](#)

[New-BrokerPowerTimeScheme](#)

[Remove-BrokerPowerTimeScheme](#)

## Parameters

**-InputObject**<PowerTimeScheme[]>

The power time scheme to be renamed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The current name of the power time scheme to be renamed.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-NewName**<String>

The new name to be applied to the power time scheme.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.PowerTimeScheme The power time scheme to be renamed.

### Return Values

None or Citrix.Broker.Admin.SDK.PowerTimeScheme

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.PowerTimeScheme object.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Rename-BrokerPowerTimeScheme -Name 'Development Weekdays' -NewName 'Dev Week'  
Renames the power time scheme named 'Development Weekdays' to 'Dev Week'.
```

# Rename-BrokerTag

Jan 04, 2017

Rename one or more tags.

## Syntax

```
Rename-BrokerTag [-InputObject] <Tag[]> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-BrokerTag [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Renames one or more tags with the supplied name.

## Related topics

[Add-BrokerTag](#)

[Get-BrokerTag](#)

[New-BrokerTag](#)

[Remove-BrokerTag](#)

## Parameters

**-InputObject**<Tag[]>

Specifies one or more tag objects to rename.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Identifies tags to be renamed by name.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-NewName**<String>

Specifies new name for the tags.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|               |                                                                                        |
|---------------|----------------------------------------------------------------------------------------|
| Required?     | false                                                                                  |
| Default Value | localhost. Once a value is provided by any cmdlet, this value will become the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

### Input Type

Citrix.Broker.Admin.SDK.Tag The tag to rename can be piped into this cmdlet.

### Return Values

None or Citrix.Broker.Admin.SDK.Tag

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.Tag object.

### Notes

Note that when renaming a tag, its UUID remains the same and any associations are maintained.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Rename-BrokerTag -Name 'OldName' -NewName 'ReplacementName'
```

Renames tags with the name 'OldName' to 'ReplacementName'.

# Reset-BrokerLicensingConnection

Jan 04, 2017

Resets the broker's license server connection.

## Syntax

```
Reset-BrokerLicensingConnection [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Reset-BrokerLicensingConnection cmdlet resets the broker's connection to the license server.

Licensing changes resulting from new license files or alterations to the site-level licensing properties don't become effective immediately. There will typically be a delay as the changes are propagated across the site based on the scheduling of refresh logic built into the controllers and the license server.

Resetting the connection causes the list of available licenses for the connection to be updated. After adding licenses or changing the site-level licensing properties you can run Reset-BrokerLicensingConnection to ensure that the broker can access the new licenses immediately.

Each broker service instance holds its own connection to the license server. In order for the licensing changes to be applied immediately throughout the XenDesktop site this command needs to be run on every controller in the site.

## Related topics

[Get-BrokerSite](#)

[Set-BrokerSite](#)

## Parameters

### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

None You cannot pipe input into this cmdlet.

### Return Values

None

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Reset-BrokerLicensingConnection  
Reset the broker's license server connection.
```



# Reset-BrokerServiceGroupMembership

Jan 04, 2017

Reloads the access permissions and configuration service locations for the Broker Service.

## Syntax

```
Reset-BrokerServiceGroupMembership -ConfigServiceInstance <PSObject[]> [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

## Detailed Description

Enables you to reload Broker Service access permissions and configuration service locations. The Reset-BrokerServiceGroupMembership command must be run on at least one instance of the service type (Broker) after installation and registration with the configuration service. Without this operation, the Broker services will be unable to communicate with other services in the XenDesktop deployment. When the command is run, the services are updated when additional services are added to the deployment, provided that the configuration service is not stopped. The Reset-BrokerServiceGroupMembership command can be run again to refresh this information if automatic updates do not occur when new services are added to the deployment. If more than one configuration service instance is passed to the command, the first instance that meets the expected service type requirements is used.

## Related topics

[Get-BrokerServiceInstance](#)

[Get-BrokerServiceStatus](#)

## Parameters

**-ConfigServiceInstance**<PSObject[]>

Specifies the configuration service instance object that represents the service instance for the type 'InterService' that references a configuration service for the deployment.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          | LocalHost      |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Sdk.ServiceInstance[] Service instances containing a ServiceInstance object that refers to the central configuration service interservice interface can be piped to the Reset-BrokerServiceGroupMembership command.

### Return Values

Citrix.Broker.Sdk.ServiceInstance

Reset-BrokerServiceGroupMembership returns opaque objects containing Configuration Service instances that are used by the Broker Service instance.

### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### NoSuitableServiceInstance

None of the supplied service instance objects were suitable for resetting service group membership.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

### Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-BrokerServiceGroupMembership
```

Reset the service group membership for a service in a deployment where the configuration service is configured and running on the same machine as the service.

----- **EXAMPLE 2** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config -AdminAddress OtherServer.example.com | Reset-BrokerServiceGroupmembership
```

Reset the service group membership for a service in a deployment where the configuration service that is configured and running on a machine named 'OtherServer.example.com'.

# Send-BrokerSessionMessage

Jan 04, 2017

Sends a message to a session.

## Syntax

```
Send-BrokerSessionMessage [-InputObject] <Session[]> [-MessageStyle] <SendMessageStyle> [-Title] <String> [-Text] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Generates a message box in the target session(s).

## Related topics

[Get-BrokerSession](#)

## Parameters

**-InputObject**<Session[]>

The target session(s) to send the message to.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-MessageStyle**<SendMessageStyle>

The style of message box to use (valid values are Critical, Question, Exclamation, or Information).

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Title**<String>

Text to display in the messagebox title bar.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                |
|------------------------|----------------|
| Accept Pipeline Input? | true (ByValue) |
|------------------------|----------------|

**-Text**<String>

The message to display.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.Session The session to which to send the message can be piped in.

**Return Values**

None

**Notes**

Sessions can be passed as the InputObject parameter as either session objects or their numeric Uids.

This operation is non-blocking and returns before it completes. The operation, however, is unlikely to fail unless there are communication problems between controller and machine, if bad arguments are passed to the cmdlet itself or if the machine cannot successfully execute the operation.

The transient nature of sessions means that the list of session objects or UIDs supplied to Send-BrokerSessionMessage could consist of valid and invalid sessions. Invalid sessions are detected and disregarded and the send message operation is invoked on the machines running valid sessions.

The system can fail to invoke the operation if the machine is not in an appropriate state or if there are problems in communicating with the machine. When an operation is invoked the system detects if the operation was initiated successfully or not by the session. As this operation is non-blocking the system doesn't detect or report whether the operation ultimately succeeded or failed after its successful initialization in the session.

Operation failures are reported through the broker SDK error handling mechanism (see about\_Broker\_ErrorHandling). In the event of errors the SdkErrorRecord error status code is set to SessionOperationFailed and its error data dictionary is populated with the following entries:

- o OperationsAttemptedCount: The number of operations attempted.
- o OperationsFailedCount - The number of failed operations.
- o OperationsSucceededCount - The number of successfully executed operations.
- o UnresolvedSessionFailuresCount - The number of operations that failed due to invalid sessions being supplied.
- o OperationInvocationFailuresCount - The number of operations that failed because they could not be invoked in the session.
- o DesktopExecutionFailuresCount - The number of operations that failed because they could not be successfully executed in the session.

The SdkErrorRecord message will also display the number of attempted, failed and successful operations in the following format:

```
"Session operation error - attempted:<OperationsAttemptedCount>, failed:<OperationsFailedCount>, succeeded:
<OperationsSucceededCount>"
```

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $sessions = Get-BrokerSession -UserName MYDOMAIN*
C:\PS> Send-BrokerSessionMessage $sessions -MessageStyle Information -Title TestTitle -Text TestMessage
Sends a message to all the sessions for any user in MYDOMAIN.
```

### ----- EXAMPLE 2 -----

```
C:\PS> $desktop = Get-BrokerDesktop -Uid 1
C:\PS> Send-BrokerSessionMessage $desktop.SessionUid -MessageStyle Information -Title TestTitle -Text TestMessage
Sends a message to the session on the desktop with Uid 1.
```

### ----- EXAMPLE 3 -----

```
C:\PS> trap [Citrix.Broker.Admin.SDK.SdkOperationException]
```

```
C:\PS> {
C:\PS> write $("Exception name = " + $_.Exception.GetType().FullName)
C:\PS> write $("SdkOperationException.Status = " + $_.Exception.Status)
C:\PS> write $("SdkOperationException.ErrorData=")
C:\PS> $_.Exception.ErrorData
C:\PS>
C:\PS> write $("SdkOperationException.InnerException = " + $_.Exception.InnerException)
C:\PS> $_.Exception.InnerException
C:\PS> continue
C:\PS> }
C:\PS>
C:\PS> Send-BrokerSessionMessage -InputObject 10,11,12 -MessageStyle Information -Title "message title" -Text "message text"
Trap and display error information.
```

# Set-BrokerAccessPolicyRule

Jan 04, 2017

Modifies an existing rule in the site's access policy.

## Syntax

```
Set-BrokerAccessPolicyRule [-InputObject] <AccessPolicyRule[]> [-PassThru] [-AddExcludedClientIPs <IPAddressRange[]>] [-AddExcludedClientNames <String[]>] [-AddExcludedSmartAccessTags <String[]>] [-AddExcludedUsers <User[]>] [-AddIncludedClientIPs <IPAddressRange[]>] [-AddIncludedClientNames <String[]>] [-AddIncludedSmartAccessTags <String[]>] [-AddIncludedUsers <User[]>] [-AllowedConnections <AllowedConnection>] [-AllowedProtocols <String[]>] [-AllowedUsers <AllowedUser>] [-AllowRestart <Boolean>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedClientIPFilterEnabled <Boolean>] [-ExcludedClientIPs <IPAddressRange[]>] [-ExcludedClientNameFilterEnabled <Boolean>] [-ExcludedClientNames <String[]>] [-ExcludedSmartAccessFilterEnabled <Boolean>] [-ExcludedSmartAccessTags <String[]>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-HdxSslEnabled <Boolean>] [-IncludedClientIPFilterEnabled <Boolean>] [-IncludedClientIPs <IPAddressRange[]>] [-IncludedClientNameFilterEnabled <Boolean>] [-IncludedClientNames <String[]>] [-IncludedSmartAccessFilterEnabled <Boolean>] [-IncludedSmartAccessTags <String[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-RemoveExcludedClientIPs <IPAddressRange[]>] [-RemoveExcludedClientNames <String[]>] [-RemoveExcludedSmartAccessTags <String[]>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedClientIPs <IPAddressRange[]>] [-RemoveIncludedClientNames <String[]>] [-RemoveIncludedSmartAccessTags <String[]>] [-RemoveIncludedUsers <User[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAccessPolicyRule [-Name] <String> [-PassThru] [-AddExcludedClientIPs <IPAddressRange[]>] [-AddExcludedClientNames <String[]>] [-AddExcludedSmartAccessTags <String[]>] [-AddExcludedUsers <User[]>] [-AddIncludedClientIPs <IPAddressRange[]>] [-AddIncludedClientNames <String[]>] [-AddIncludedSmartAccessTags <String[]>] [-AddIncludedUsers <User[]>] [-AllowedConnections <AllowedConnection>] [-AllowedProtocols <String[]>] [-AllowedUsers <AllowedUser>] [-AllowRestart <Boolean>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedClientIPFilterEnabled <Boolean>] [-ExcludedClientIPs <IPAddressRange[]>] [-ExcludedClientNameFilterEnabled <Boolean>] [-ExcludedClientNames <String[]>] [-ExcludedSmartAccessFilterEnabled <Boolean>] [-ExcludedSmartAccessTags <String[]>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-HdxSslEnabled <Boolean>] [-IncludedClientIPFilterEnabled <Boolean>] [-IncludedClientIPs <IPAddressRange[]>] [-IncludedClientNameFilterEnabled <Boolean>] [-IncludedClientNames <String[]>] [-IncludedSmartAccessFilterEnabled <Boolean>] [-IncludedSmartAccessTags <String[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-RemoveExcludedClientIPs <IPAddressRange[]>] [-RemoveExcludedClientNames <String[]>] [-RemoveExcludedSmartAccessTags <String[]>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedClientIPs <IPAddressRange[]>] [-RemoveIncludedClientNames <String[]>] [-RemoveIncludedSmartAccessTags <String[]>] [-RemoveIncludedUsers <User[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerAccessPolicyRule cmdlet modifies an existing rule in the site's access policy.

An access policy rule defines a set of connection filters and access control rights relating to a desktop group. These allow fine-grained control of what access is granted to a desktop group based on details of, for example, a user's endpoint device, its address, and the user's identity.

Changing a rule does not affect existing user sessions, but it may result in users being unable to launch new sessions, or reconnect to disconnected sessions if the change removes access to the desktop group delivering those sessions.

## Related topics

[New-BrokerAccessPolicyRule](#)

[Get-BrokerAccessPolicyRule](#)

[Rename-BrokerAccessPolicyRule](#)

[Remove-BrokerAccessPolicyRule](#)

## Parameters

**-InputObject** <AccessPolicyRule[]>

The access policy rule to be modified.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|



|                        |                |
|------------------------|----------------|
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The name of the access policy rule to be modified.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-AddExcludedClientIPs**<IPAddressRange[]>

Adds the specified user device IP addresses to the excluded client IP address filter of the rule.

See the ExcludedClientIPs parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AddExcludedClientNames**<String[]>

Adds the specified user device names to the excluded client names filter of the rule.

See the ExcludedClientNames parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AddExcludedSmartAccessTags**<String[]>

Adds the specified SmartAccess tags to the excluded SmartAccess tags filter of the rule.

See the ExcludedSmartAccessTags parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AddExcludedUsers**<User[]>

Adds the specified users and groups to the excluded users filter of the rule.

See the ExcludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AddIncludedClientIPs**<IPAddressRange[]>

Adds the specified user device IP addresses to the included client IP address filter of the rule.

See the IncludedClientIPs parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AddIncludedClientNames**<String[]>

Adds the specified user device names to the included client names filter of the rule.

See the IncludedClientNames parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AddIncludedSmartAccessTags**<String[]>

Adds the specified SmartAccess tags to the included SmartAccess tags filter of the rule.

See the IncludedSmartAccessTags parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-AddIncludedUsers**<User[]>

Adds the specified users and groups to the included users filter of the rule.

See the IncludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AllowedConnections**<AllowedConnection>

Changes whether connections must be local or via Access Gateway, and if so whether specified SmartAccess tags must be provided by Access Gateway with the connection. This property forms part of the included SmartAccess tags filter.

Valid values are Filtered, NotViaAG, and ViaAG.

For a detailed description of this property see "help about\_Broker\_AccessPolicy".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AllowedProtocols**<String[]>

Changes the protocols (for example HDX, RDP) available to the user for sessions delivered from the rule's desktop group. If the user gains access to a desktop group by multiple rules, the allowed protocol list is the combination of the protocol lists from all those rules.

If the protocol list is empty, access to the desktop group is implicitly denied.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AllowedUsers**<AllowedUser>

Changes the behavior of the included users filter of the rule. This can restrict access to a list of named users or groups, allow access to any authenticated user, any user (whether authenticated or not), or only non-authenticated users. For a detailed description of this property see "help about\_Broker\_AccessPolicy".

Valid values are Filtered, AnyAuthenticated, Any, AnonymousOnly and FilteredOrAnonymous.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AllowRestart**<Boolean>

Changes whether the user can restart sessions delivered from the rule's desktop group. Session restart is handled as follows: For sessions on single-session power-managed machines, the machine is powered off, and a new session launch request made; for sessions on multi-session machines, a logoff request is issued to the session, and a new session launch request made; otherwise the property is ignored.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Changes the description of the rule. The text is purely informational for the administrator, it is never visible to the end user.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Changes whether the rule is enabled or disabled. A disabled rule is ignored when evaluating the site's access policy.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedClientIPFilterEnabled**<Boolean>

Changes whether the excluded client IP address filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedClientIPs**<IPAddressRange[]>

Changes the IP addresses of user devices explicitly denied access to the rule's desktop group. Addresses can be specified as simple numeric addresses or as subnet masks (for example, 10.40.37.5 or 10.40.0.0/16). This property forms part of the excluded client IP address filter.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ExcludedClientNameFilterEnabled**<Boolean>

Changes whether the excluded client names filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedClientNames**<String[]>

Changes which names of user devices are explicitly denied access to the rule's desktop group. This property forms part of the excluded client names filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedSmartAccessFilterEnabled**<Boolean>

Changes whether the excluded SmartAccess tags filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedSmartAccessTags**<String[]>

Changes which SmartAccess tags explicitly deny access to the rule's desktop group if any occur in those provided by Access Gateway with the user's connection. This property forms part of the excluded SmartAccess tags filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUserFilterEnabled**<Boolean>

Changes whether the excluded users filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ExcludedUsers**<User[]>

Changes which users and groups are explicitly denied access to the rule's desktop group. This property forms part of the excluded users filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HdxSslEnabled**<Boolean>

Indicates whether SSL encryption is enabled for sessions delivered from the rule's desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedClientIPFilterEnabled**<Boolean>

Changes whether the included client IP address filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedClientIPs**<IPAddressRange[]>

Changes which IP addresses of user devices allowed access to the rule's desktop group. Addresses can be specified as simple numeric addresses or as subnet masks (for example, 10.40.37.5 or 10.40.0.0/16). This property forms part of the included client IP address filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedClientNameFilterEnabled**<Boolean>

Changes whether the included client name filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-IncludedClientNames<String[]>**

Changes which names of user devices are allowed access to the rule's desktop group. This property forms part of the included client names filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedSmartAccessFilterEnabled<Boolean>**

Changes whether the included SmartAccess tags filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedSmartAccessTags<String[]>**

Changes which SmartAccess tags grant access to the rule's desktop group if any occur in those provided by Access Gateway with the user's connection. This property forms part of the excluded SmartAccess tags filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUserFilterEnabled<Boolean>**

Changes whether the included users filter is enabled or disabled. If the filter is disabled, it is ignored when the access policy rule is evaluated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUsers<User[]>**

Changes which users and groups are granted access to the rule's desktop group. This property forms part of the included users filter.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-RemoveExcludedClientIPs**<IPAddressRange[]>

Removes the specified user device IP addresses from the excluded client IP address filter of the rule.

See the ExcludedClientIPs parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveExcludedClientNames**<String[]>

Removes the specified user device names from the excluded client names filter of the rule.

See the ExcludedClientNames parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveExcludedSmartAccessTags**<String[]>

Removes the specified SmartAccess tags from the excluded SmartAccess tags filter of the rule.

See the ExcludedSmartAccessTags parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveExcludedUsers**<User[]>

Removes the specified users and groups from the excluded users filter of the rule.

See the ExcludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveIncludedClientIPs**<IPAddressRange[]>

Removes the specified user device IP addresses from the included client IP address filter of the rule.



See the IncludedClientIPs parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveIncludedClientNames<String[]>**

Removes the specified client names from the included client names filter of the rule.

See the IncludedClientNames parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveIncludedSmartAccessTags<String[]>**

Removes the specified SmartAccess tags from the included SmartAccess tags filter of the rule.

See the IncludedSmartAccessTags parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveIncludedUsers<User[]>**

Removes the specified users and groups from the included users filter of the rule.

See the IncludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.AccessPolicyRule The access policy rule to be modified.

**Return Values**

None, or Citrix.Broker.Admin.SDK.AccessPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AccessPolicyRule object.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Set-BrokerAccessPolicyRule 'Temp Staff' -AddIncludedUsers office\contractors

Adds user group OFFICE\contractors to the Temp Staff access policy rule. The resources that the group can access are dependent on the existing properties of the rule in addition to the site's assignment and entitlement policies.

----- **EXAMPLE 2** -----

C:\PS> Set-BrokerAccessPolicyRule 'Temp Staff' -ExcludedClientIPFilterEnabled \$true -AddExcludedClientIPs '10.15.0.0/16' -AllowedConnections ViaAG

Modifies the Temp Staff access policy rule to remove access to any user device with an IP address matching 10.15.0.0/16, and requires that all connections by the rule must come through Access Gateway (assuming that the included SmartAccess tags filter is enabled).

# Set-BrokerAccessPolicyRuleMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for AccessPolicyRule

## Syntax

```
Set-BrokerAccessPolicyRuleMetadata [-AccessPolicyRuleId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAccessPolicyRuleMetadata [-AccessPolicyRuleId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAccessPolicyRuleMetadata [-AccessPolicyRuleId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAccessPolicyRuleMetadata [-InputObject] <AccessPolicyRule[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAccessPolicyRuleMetadata [-InputObject] <AccessPolicyRule[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAccessPolicyRuleMetadata [-AccessPolicyRuleName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAccessPolicyRuleMetadata [-AccessPolicyRuleName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerAccessPolicyRuleMetadata cmdlet creates/updates metadata key-value pairs for AccessPolicyRule. The AccessPolicyRule can be specified by InputObject or piping.

## Related topics

### Parameters

**-AccessPolicyRuleId**<Int32>

Specifies the AccessPolicyRule object whose Metadata is to be created/updated by ID.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<AccessPolicyRule[]>

Specifies the AccessPolicyRule objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AccessPolicyRuleName**<String>

Specifies the AccessPolicyRule object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerAccessPolicyRule You can pipe the AccessPolicyRule to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerAccessPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerAccessPolicyRule object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerAccessPolicyRuleMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the AccessPolicyRule whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerAccessPolicyRule | Set-BrokerAccessPolicyRuleMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the AccessPolicyRule in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerAccessPolicyRuleMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the AccessPolicyRule in the site whose name is 'objname'

# Set-BrokerAdminFolderMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for AdminFolder

## Syntax

```
Set-BrokerAdminFolderMetadata [-AdminFolderId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAdminFolderMetadata [-AdminFolderId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAdminFolderMetadata [-AdminFolderId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAdminFolderMetadata [-InputObject] <AdminFolder[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAdminFolderMetadata [-InputObject] <AdminFolder[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAdminFolderMetadata [-AdminFolderName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAdminFolderMetadata [-AdminFolderName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerAdminFolderMetadata cmdlet creates/updates metadata key-value pairs for AdminFolder. The AdminFolder can be specified by InputObject or piping.

## Related topics

### Parameters

**-AdminFolderId**<Int32>

Specifies the AdminFolder object whose Metadata is to be created/updated by ID.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<AdminFolder[]>

Specifies the AdminFolder objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AdminFolderName**<String>

Specifies the AdminFolder object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated



|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerAdminFolder You can pipe the AdminFolder to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerAdminFolder

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerAdminFolder object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerAdminFolderMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the AdminFolder whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerAdminFolder | Set-BrokerAdminFolderMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the AdminFolder in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerAdminFolderMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the AdminFolder in the site whose name is 'objname'

# Set-BrokerAppAssignmentPolicyRule

Jan 04, 2017

Modifies an existing application rule in the site's assignment policy.

## Syntax

```
Set-BrokerAppAssignmentPolicyRule [-InputObject] <AppAssignmentPolicyRule[]> [-PassThru] [-AddExcludedUsers <User[]>] [-AddIncludedUsers <User[]>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedUsers <User[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAppAssignmentPolicyRule [-Name] <String> [-PassThru] [-AddExcludedUsers <User[]>] [-AddIncludedUsers <User[]>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedUsers <User[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerAppAssignmentPolicyRule cmdlet modifies an existing application rule in the site's assignment policy.

An application rule in the assignment policy defines the users who are entitled to a self-service persistent machine assignment from the rule's desktop group; once assigned the machine can run one or more applications published from the group.

Changing an application rule does not alter machine assignments that have already been made by the rule, nor does it affect active sessions to those machines in any way.

## Related topics

[New-BrokerAppAssignmentPolicyRule](#)

[Get-BrokerAppAssignmentPolicyRule](#)

[Rename-BrokerAppAssignmentPolicyRule](#)

[Remove-BrokerAppAssignmentPolicyRule](#)

## Parameters

**-InputObject** <AppAssignmentPolicyRule[]>

The application rule in the assignment policy to be modified.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the application rule in the assignment policy to be modified.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-AddExcludedUsers**<User[]>

Adds the specified users to the excluded users filter of the application rule, that is, the users and groups who are explicitly denied an entitlement to a machine assignment from this rule.

See the ExcludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AddIncludedUsers**<User[]>

Adds the specified users to the included users filter of the application rule, that is, the users and groups who are granted an entitlement to a machine assignment by the rule.

See the IncludedUsers parameter for more information.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Changes the description of the application rule. The text is purely informational for the administrator, it is never visible to the end user.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Enables or disables the application rule. A disabled rule is ignored when evaluating the site's assignment policy.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUserFilterEnabled**<Boolean>

Enables or disables the excluded users filter. If the filter is disabled then any user entries in the filter are ignored when assignment policy rules are evaluated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUsers**<User[]>

Changes the excluded users filter of the application rule, that is, the users and groups who are explicitly denied an entitlement to a machine assignment from the rule.

This can be used to exclude users or groups who would otherwise gain access by groups specified in the included users filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUserFilterEnabled**<Boolean>

Enables or disables the included users filter. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to a machine assignment by the application rule.

Users who would be implicitly granted access when the filter is disabled can still be explicitly denied access using the excluded users filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUsers**<User[]>

Changes the included users filter of the application rule, that is, the users and groups who are granted an entitlement to a machine assignment by the rule.

If a user appears explicitly in the excluded users filter of the rule, or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveExcludedUsers**<User[]>

Removes the specified users from the excluded users filter of the application rule, that is, the users and groups who are explicitly denied an entitlement to a machine assignment from this rule.

See the ExcludedUsers parameter for more information.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveIncludedUsers<User[]>**

Removes the specified users from the included users filter of the application rule, that is, the users and groups who are granted an entitlement to a machine assignment by the rule.

See the IncludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.AppAssignmentPolicyRule The application rule within the assignment policy to be modified.

## Return Values

None or Citrix.Broker.Admin.SDK.AppAssignmentPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AppAssignmentPolicyRule object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerAppAssignmentPolicyRule 'Temp Staff' -AddIncludedUsers office\interns
```

Adds the user group OFFICE\interns to the Temp Staff application rule in the assignment policy. This grants all members of that user group an entitlement to a machine in the rule's desktop group. The machines can run applications published from the group. The application session properties obtained using the rule are determined by the rule's other properties.

### ----- EXAMPLE 2 -----

```
C:\PS> Set-BrokerAppAssignmentPolicyRule 'Temp Staff' -Enabled $false
```

Disables the Temp Staff application rule in the assignment policy. This prevents further machine assignments being made using this rule until it is re-enabled. However, access to machines already assigned using the rule is not impacted.



# Set-BrokerAppEntitlementPolicyRule

Jan 04, 2017

Modifies an existing application rule in the site's entitlement policy.

## Syntax

```
Set-BrokerAppEntitlementPolicyRule [-InputObject] <AppEntitlementPolicyRule[]> [-PassThru] [-AddExcludedUsers <User[]>] [-AddIncludedUsers <User[]>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedUsers <User[]>] [-SessionReconnection <SessionReconnection>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAppEntitlementPolicyRule [-Name] <String> [-PassThru] [-AddExcludedUsers <User[]>] [-AddIncludedUsers <User[]>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedUsers <User[]>] [-SessionReconnection <SessionReconnection>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerAppEntitlementPolicyRule cmdlet modifies an existing application rule in the site's entitlement policy.

An application rule in the entitlement policy defines the users who are allowed per-session access to a machine to run one or more applications published from the rule's desktop group.

Changing a rule does not affect existing sessions launched using the rule, but if the change removes an entitlement to a machine that was previously granted, users may be unable to reconnect to a disconnected session on that machine.

## Related topics

[New-BrokerAppEntitlementPolicyRule](#)

[Get-BrokerAppEntitlementPolicyRule](#)

[Rename-BrokerAppEntitlementPolicyRule](#)

[Remove-BrokerAppEntitlementPolicyRule](#)

## Parameters

**-InputObject** <AppEntitlementPolicyRule[]>

The application rule in the entitlement policy to be modified.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The name of the application rule in the entitlement policy to be modified.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-AddExcludedUsers**<User[]>

Adds the specified users to the excluded users filter of the rule, that is, the users and groups who are explicitly denied entitlements to run applications published from the desktop group.

See the ExcludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AddIncludedUsers**<User[]>

Adds the specified users to the included users filter of the rule, that is, the users and groups who are granted an entitlement to an application session by the rule.

See the IncludedUsers parameter for more information.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Changes the description of the application rule. The text is purely informational for the administrator, it is never visible to the end user.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Enables or disables the application rule. A disabled rule is ignored when evaluating the site's entitlement policy.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUserFilterEnabled**<Boolean>

Enables or disables the excluded users filter. If the filter is disabled then any user entries in the filter are ignored when entitlement policy rules are evaluated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUsers**<User[]>

Changes the excluded users filter of the rule, that is, the users and groups who are explicitly denied entitlements to run applications published from the desktop group.

This can be used to exclude users or groups or users who would otherwise gain access by groups specified in the included

users filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUserFilterEnabled**<Boolean>

Enables or disables the included users filter. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to an application session by the application rule.

Users who would be implicitly granted access when the filter is disabled can still be explicitly denied access using the excluded users filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUsers**<User[]>

Changes the included users filter of the rule, that is, the users and groups who are granted an entitlement to an application session by the rule.

If a user appears explicitly in the excluded users filter of the rule or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveExcludedUsers**<User[]>

Removes the specified users from the excluded users filter of the application rule, that is, the users and groups who are explicitly denied entitlements to run applications published from the desktop group.

See the ExcludedUsers parameter for more information.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveIncludedUsers**<User[]>

Removes the specified users from the included users filter of the rule, that is, the users and groups who are granted an entitlement to an application session by the rule.

See the IncludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionReconnection**<SessionReconnection>

Defines reconnection (roaming) behavior for sessions launched using this rule. Session reconnection control is an experimental and unsupported feature.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule The application rule in the entitlement policy rule to be modified.

### Return Values

None or Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AppEntitlementPolicyRule object.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerAppEntitlementPolicyRule 'Temp Workers' -AddIncludedUsers office\contractors
```

Adds the user group OFFICE\contractors to those entitled to run applications from the rule's associated desktop group. This grants all members of that group an entitlement to an application session from that group.

----- **EXAMPLE 2** -----

```
C:\PS> Set-BrokerAppEntitlementPolicyRule 'Temp Workers' -Enabled $false
```

Disables the Temp Workers application rule in the entitlement policy. This prevents further application sessions being launched using this rule until it is re-enabled. However, access to existing application sessions is not affected.

# Set-BrokerApplication

Jan 04, 2017

Changes the settings of an application to the value specified in the command.

## Syntax

```
Set-BrokerApplication [-InputObject] <Application[]> [-PassThru] [-BrowserName <String>] [-ClientFolder <String>] [-CommandLineArguments <String>] [-CommandLineExecutable <String>] [-CpuPriorityLevel <CpuPriorityLevel>] [-Description <String>] [-Enabled <Boolean>] [-IconFromClient <Boolean>] [-IconUid <Int32>] [-PublishedName <String>] [-SecureCmdLineArgumentsEnabled <Boolean>] [-ShortcutAddedToDesktop <Boolean>] [-ShortcutAddedToStartMenu <Boolean>] [-StartMenuFolder <String>] [-UserFilterEnabled <Boolean>] [-Visible <Boolean>] [-WaitForPrinterCreation <Boolean>] [-WorkingDirectory <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplication [-Name] <String> [-PassThru] [-BrowserName <String>] [-ClientFolder <String>] [-CommandLineArguments <String>] [-CommandLineExecutable <String>] [-CpuPriorityLevel <CpuPriorityLevel>] [-Description <String>] [-Enabled <Boolean>] [-IconFromClient <Boolean>] [-IconUid <Int32>] [-PublishedName <String>] [-SecureCmdLineArgumentsEnabled <Boolean>] [-ShortcutAddedToDesktop <Boolean>] [-ShortcutAddedToStartMenu <Boolean>] [-StartMenuFolder <String>] [-UserFilterEnabled <Boolean>] [-Visible <Boolean>] [-WaitForPrinterCreation <Boolean>] [-WorkingDirectory <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerApplication cmdlet changes the value of one or more properties of an application, such as its CpuPriorityLevel or its CommandLineArguments, to the value specified in the command.

This cmdlet lets you change only the settings of the Application object, and not the relationships to other objects. For instance, it does not let you change which users can access this application, or change which desktop groups this application is published to. To do this, you need to remove the existing association, and then add a new association. The following example shows how to change the desktop group that an application is associated with from \$group1 to \$group2:

```
Remove-BrokerApplication -DesktopGroup $group1
```

```
Add-RemoveApplication -DesktopGroup $group2
```

You can change properties of both HostedOnDesktop and InstalledOnClient applications but it is not possible to change the ApplicationType. Also, the Name cannot be changed using this cmdlet; to do this, use the Rename-BrokerApplication cmdlet.

## Related topics

[New-BrokerApplication](#)

[Add-BrokerApplication](#)

[Get-BrokerApplication](#)

[Remove-BrokerApplication](#)

[Rename-BrokerApplication](#)

## Move-BrokerApplication

### Parameters

**-InputObject**<Application[]>

Specifies the application to modify. The Uid of the application can also be substituted for the object.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the application to be modified.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-BrowserName**<String>

Specifies the name of the application to modify. Note that the BrowserName cannot be changed in this manner; to modify the BrowserName of an application, use the Rename-BrokerApplication cmdlet.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |



|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Client Folder**<String>

Specifies the folder that the application belongs to as the user sees it. This is the application folder displayed in the Citrix Online Plug-in, in Web Services, and also in the user's start menu. Subdirectories can be specified with '\' character. The following special characters are not allowed: / \* ? < > | " :. Note that this property cannot be set for applications of type InstalledOnClient.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CommandLineArguments**<String>

Specifies the command-line arguments to use when launching the executable.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CommandLineExecutable**<String>

Specifies the name of the executable file to launch.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-CpuPriorityLevel**<CpuPriorityLevel>

Specifies the CPU priority for the launched executable. Valid values are: Low, BelowNormal, Normal, AboveNormal, and High. Note that this property cannot be set for applications of type InstalledOnClient.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Specifies the description of the application.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Specifies whether or not this application can be launched.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconFromClient**<Boolean>

Specifies if the app icon should be retrieved from the application on the client. This is reserved for possible future use, and all applications of type HostedOnDesktop cannot set or change this value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconUid**<Int32>

Specifies which icon to use for this application. This application is visible both to the administrator (in the consoles) and also to the user. If no icon is specified, then a generic built-in application icon is used.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName**<String>

Specifies the name seen by end users who have access to this application.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureCmdLineArgumentsEnabled**<Boolean>

Specifies whether the command-line arguments should be secured. This is reserved for possible future use, and all applications of type HostedOnDesktop can only have this value set to true.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ShortcutAddedToDesktop**<Boolean>

Specifies whether or not a shortcut to the application should be placed on the user device.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ShortcutAddedToStart Menu**<Boolean>

Specifies whether a shortcut to the application should be placed in the user's start menu on their user device.

|           |       |
|-----------|-------|
| Required? | false |
|           |       |

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartMenuFolder**<String>

Specifies the name of the start menu folder that holds the application shortcut (if any). This is valid only for the Citrix Online Plug-in. Subdirectories can be specified with '\' character. The following special characters are not allowed: / \* ? < > | " .:

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserFilterEnabled**<Boolean>

Specifies whether the application's user filter is enabled or disabled. Where the user filter is enabled, the application is only visible to users who appear in the filter (either explicitly or by virtue of group membership).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Visible**<Boolean>

Specifies whether or not this application is visible to users. Note that it's possible for an application to be disabled and still visible.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WaitForPrinterCreation**<Boolean>

Specifies whether or not the session waits for the printers to be created before allowing the end-user to interact with the session. Note that this property cannot be set for applications of type InstalledOnClient.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WorkingDirectory**<String>

Specifies from which working directory the executable is launched from.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.Application, or depends on parameter You can pipe the application to be added to Set-

BrokerApplication. You can also pipe some of the other parameters by name.

## Return Values

None or Citrix.Broker.Admin.SDK.Application

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.Application object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerApplication -Name "Notepad" -Description 'Windows Notepad'
```

Modifies the application that has a Name of "Notepad" so that its description reads Windows Notepad.

### ----- EXAMPLE 2 -----

```
C:\PS> $app = Get-BrokerApplication -BrowserName "Calculator"
```

```
C:\PS> Set-BrokerApplication -InputObject $app -Enabled $false
```

First gets the application with a BrowserName of "Calculator", then modifies that application (by supplying the application object in the first position) so that it is disabled for users.

# Set-BrokerApplicationInstanceMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for ApplicationInstance

## Syntax

```
Set-BrokerApplicationInstanceMetadata [-ApplicationInstanceId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationInstanceMetadata [-ApplicationInstanceId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationInstanceMetadata [-ApplicationInstanceId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationInstanceMetadata [-InputObject] <ApplicationInstance[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationInstanceMetadata [-InputObject] <ApplicationInstance[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerApplicationInstanceMetadata cmdlet creates/updates metadata key-value pairs for ApplicationInstance. The ApplicationInstance can be specified by InputObject or piping.

## Related topics

### Parameters

#### **-ApplicationInstanceId**<Int64>

Specifies the ApplicationInstance object whose Metadata is to be created/updated by ID.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

#### **-InputObject**<ApplicationInstance[]>

Specifies the ApplicationInstance objects whose Metadata is to be created/updated.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                |
|------------------------|----------------|
| Accept Pipeline Input? | true (ByValue) |
|------------------------|----------------|

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | False |



|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.BrokerApplicationInstance You can pipe the ApplicationInstance to hold the new or updated metadata.

**Return Values**

None or Citrix.Broker.Admin.SDK.BrokerApplicationInstance

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerApplicationInstance object.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Set-BrokerApplicationInstanceMetadata -InputObject \$obj-Uid -Name "MyMetadataName" -Value "1234"  
 This command creates/updates the Metadata "MyMetadataName" key-value pair for the ApplicationInstance whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerApplicationInstance | Set-BrokerApplicationInstanceMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the ApplicationInstance in the site

# Set-BrokerApplicationMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for Application

## Syntax

```
Set-BrokerApplicationMetadata [-ApplicationId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-ApplicationId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-ApplicationId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-InputObject] <Application[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-InputObject] <Application[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-ApplicationName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-ApplicationName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerApplicationMetadata cmdlet creates/updates metadata key-value pairs for Application. The Application can be specified by InputObject or piping.

## Related topics

### Parameters

**-ApplicationId**<Int32>

Specifies the Application object whose Metadata is to be created/updated by ID.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-InputObject**<Application[]>

Specifies the Application objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-ApplicationName**<String>

Specifies the Application object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerApplication You can pipe the Application to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerApplication

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerApplication object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerApplicationMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Application whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerApplication | Set-BrokerApplicationMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Application in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerApplicationMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Application in the site whose name is 'objname'

# Set-BrokerAssignmentPolicyRule

Jan 04, 2017

Modifies an existing desktop rule in the site's assignment policy.

## Syntax

```
Set-BrokerAssignmentPolicyRule [-InputObject] <AssignmentPolicyRule[]> [-PassThru] [-AddExcludedUsers <User[]>] [-AddIncludedUsers <User[]>] [-ColorDepth <ColorDepth>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IconUid <Int32>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-MaxDesktops <Int32>] [-PublishedName <String>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedUsers <User[]>] [-SecureIcaRequired <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRule [-Name] <String> [-PassThru] [-AddExcludedUsers <User[]>] [-AddIncludedUsers <User[]>] [-ColorDepth <ColorDepth>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IconUid <Int32>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-MaxDesktops <Int32>] [-PublishedName <String>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedUsers <User[]>] [-SecureIcaRequired <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerAssignmentPolicyRule cmdlet modifies an existing desktop rule in the site's assignment policy.

A desktop rule in the assignment policy defines the users who are entitled to self-service persistent machine assignments from the rule's desktop group. A rule defines how many machines a user is allowed from the group for delivery of full desktop sessions.

Changing a desktop rule does not alter machine assignments that have already been made by the rule, nor does it affect active sessions to those machines in any way.

## Related topics

[New-BrokerAssignmentPolicyRule](#)

[Get-BrokerAssignmentPolicyRule](#)

[Rename-BrokerAssignmentPolicyRule](#)

[Remove-BrokerAssignmentPolicyRule](#)

## Parameters

**-InputObject**<AssignmentPolicyRule[]>

The desktop rule in the assignment policy to be modified.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                |
|------------------------|----------------|
| Accept Pipeline Input? | true (ByValue) |
|------------------------|----------------|

**-Name**<String>

Specifies the name of the desktop rule in the assignment policy to be modified.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-AddExcludedUsers**<User[]>

Adds the specified users to the excluded users filter of the rule, that is, the users and groups who are explicitly denied an entitlement to a machine assignment from this rule.

See the ExcludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AddIncludedUsers**<User[]>

Adds the specified users to the included users filter of the rule, that is, the users and groups who are granted an entitlement to a machine assignment by the rule.

See the IncludedUsers parameter for more information.



|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ColorDepth**<ColorDepth>

Changes the color depth of any desktop sessions to machines assigned by the rule.

Valid values are \$null, FourBit, EightBit, SixteenBit, and TwentyFourBit.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Changes the description of the desktop rule. The text may be visible to the end user, for example, as a tooltip associated with the desktop entitlement.

A null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Enables or disables the desktop rule. A disabled rule is ignored when evaluating the site's assignment policy.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUserFilterEnabled**<Boolean>

Enables or disables the excluded users filter. If the filter is disabled then any user entries in the filter are ignored when assignment policy rules are evaluated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUsers**<User[]>

Changes the excluded users filter of the desktop rule, that is, the users and groups who are explicitly denied an entitlement to a machine assignment from this rule.

This can be used to exclude users or groups who would otherwise gain access by groups specified in the included users filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconUid**<Int32>

Changes the unique ID of the icon used to display the machine assignment entitlement to the user, and of the assigned desktop itself following the assignment.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUserFilterEnabled**<Boolean>

Enables or disables the included users filter. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to a machine assignment by the rule.

Users who would be implicitly granted access when the filter is disabled can still be explicitly denied access using the excluded users filter.

For rules that relate to RemotePC desktop groups however, if the included user filter is disabled, the rule is effectively disabled.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUsers**<User[]>

Changes the included users filter of the rule, that is, the users and groups who are granted an entitlement to a machine assignment by the rule.

If a user appears explicitly in the excluded users filter of the rule, or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxDesktops**<Int32>

The number of machines from the rule's desktop group to which a user is entitled. Where an entitlement is granted to a user group rather than an individual, the number of machines applies to each member of the user group independently.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName**<String>

Changes the published name of the machine assignment entitlement as seen by the user. Changing the published name does not affect the names of desktops that have already been assigned by the rule.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveExcludedUsers**<User[]>

Removes the specified users from the excluded users filter of the rule, that is, the users and groups who are explicitly denied an entitlement to a machine assignment from this rule.

See the ExcludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveIncludedUsers**<User[]>

Removes the specified users from the included users filter of the desktop rule, that is, the users and groups who are granted an entitlement to a machine assignment by the rule.

See the IncludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaRequired**<Boolean>

Changes whether the desktop rule requires the SecureICA protocol for desktop sessions to machines assigned using the entitlement.

The default null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.AssignmentPolicyRule The desktop rule within the assignment policy to be modified.

### Return Values

None or Citrix.Broker.Admin.SDK.AssignmentPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.AssignmentPolicyRule object.

### Examples

#### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerAssignmentPolicyRule 'Temp Staff' -AddIncludedUsers office\interns
```

Adds the user group OFFICE\interns to the Temp Staff desktop rule in the assignment policy. This grants all members of that user group entitlements to machines in the rule's desktop group. The number of machine entitlements per user and the session properties of the desktops obtained using the rule are determined by the rule's other properties.

#### ----- EXAMPLE 2 -----

```
C:\PS> Set-BrokerAssignmentPolicyRule 'Temp Staff' -Enabled $false
```

Disables the Temp Staff desktop rule in the assignment policy. This prevents further machine assignments being made using

this rule until it is re-enabled. However, access to machines already assigned using the rule is not impacted.

# Set-BrokerAssignmentPolicyRuleMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for AssignmentPolicyRule

## Syntax

```
Set-BrokerAssignmentPolicyRuleMetadata [-AssignmentPolicyRuleId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-AssignmentPolicyRuleId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-AssignmentPolicyRuleId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-InputObject] <AssignmentPolicyRule[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-InputObject] <AssignmentPolicyRule[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-AssignmentPolicyRuleName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-AssignmentPolicyRuleName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerAssignmentPolicyRuleMetadata cmdlet creates/updates metadata key-value pairs for AssignmentPolicyRule. The AssignmentPolicyRule can be specified by InputObject or piping.

## Related topics

### Parameters

**-AssignmentPolicyRuleId**<Int32>

Specifies the AssignmentPolicyRule object whose Metadata is to be created/updated by ID.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-InputObject**<AssignmentPolicyRule[]>

Specifies the AssignmentPolicyRule objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AssignmentPolicyRuleName**<String>

Specifies the AssignmentPolicyRule object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|           |      |
|-----------|------|
| Required? | true |
|           |      |



|                        |                |
|------------------------|----------------|
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.BrokerAssignmentPolicyRule You can pipe the AssignmentPolicyRule to hold the new or updated metadata.

**Return Values**

None or Citrix.Broker.Admin.SDK.BrokerAssignmentPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerAssignmentPolicyRule object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerAssignmentPolicyRuleMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the AssignmentPolicyRule whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerAssignmentPolicyRule | Set-BrokerAssignmentPolicyRuleMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the AssignmentPolicyRule in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerAssignmentPolicyRuleMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the AssignmentPolicyRule in the site whose name is 'objname'

# Set-BrokerCatalog

Jan 04, 2017

Sets the properties of a catalog.

## Syntax

```
Set-BrokerCatalog [-InputObject] <Catalog[]> [-PassThru] [-Description <String>] [-IsRemotePC <Boolean>] [-MinimumFunctionalLevel <FunctionalLevel>] [-ProvisioningSchemeld <Guid>] [-PvsAddress <String>] [-PvsDomain <String>] [-RemotePCHypervisorConnectionUid <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerCatalog [-Name] <String> [-PassThru] [-Description <String>] [-IsRemotePC <Boolean>] [-MinimumFunctionalLevel <FunctionalLevel>] [-ProvisioningSchemeld <Guid>] [-PvsAddress <String>] [-PvsDomain <String>] [-RemotePCHypervisorConnectionUid <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerCatalog cmdlet sets properties of a catalog or set of catalogs. The catalog can be specified by name, in which case only one catalog can be specified, or one or more catalog instances can be passed to the command either by piping or by using the -InputObject parameter.

## Related topics

[Get-BrokerCatalog](#)

[New-BrokerCatalog](#)

[Remove-BrokerCatalog](#)

[Rename-BrokerCatalog](#)

## Parameters

**-InputObject**<Catalog[]>

Specifies the catalog objects to modify.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Identifies the catalog to modify.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-Description**<String>

Supplies the new value of the Description property.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsRemotePC**<Boolean>

Supplies a new value for IsRemotePC.

IsRemotePC can only be enabled when:

- o SessionSupport is SingleSession
- o MachinesArePhysical is true.

IsRemotePC can only be set from true to false when no RemotePCAccount references this catalog, and when no Remote PC relationship exists between this catalog and a desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MinimumFunctionalLevel**<FunctionalLevel>

The new minimum FunctionalLevel required for machines to work successfully in the catalog. If this is higher than the FunctionalLevel of any machines already in the catalog, they will immediately cease to function.

Valid values are L5, L7, L7\_6

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ProvisioningSchemeId**<Guid>

Specifies the identity of the MCS provisioning scheme the catalog is associated with (can only be specified for new catalogs with a ProvisioningType of MCS; once set can never be changed).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvsAddress**<String>

Supplies the new value of the PvsAddress property. Can only be set if CatalogKind is Pvs or PvsPvd.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PvsDomain**<String>

Supplies the new value of the PvsDomain property. Can only be set if CatalogKind is PvsPvd.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

|  |  |
|--|--|
|  |  |
|--|--|

**-RemotePCHypervisorConnectionUid<Int32>**

Supplies the new hypervisor connection to use for powering on remote PCs in this catalog (only allowed when IsRemotePC is true); this will affect all machines already in the catalog as well as those created later.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.Catalog You can pipe the catalogs to be modified to Set-BrokerCatalog.

**Return Values**

None or Citrix.Broker.Admin.SDK.Catalog

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a

Citrix.Broker.Admin.SDK.Catalog object.

## Notes

A catalog's Name property cannot be changed by Set-BrokerCatalog. To rename a catalog use Rename-BrokerCatalog.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerCatalog -Name "MyCatalog" -Description "New Description"
```

This example specifies a catalog by name and sets its description.

### ----- EXAMPLE 2 -----

```
C:\PS> $permCatalogs = Get-BrokerCatalog -AllocationType Static
```

```
C:\PS> Set-BrokerCatalog -InputObject $permCatalogs -Description "Permanently assigned machines"
```

This example sets the description for all catalogs with AllocationType 'Static'.

# Set-BrokerCatalogMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for Catalog

## Syntax

```
Set-BrokerCatalogMetadata [-CatalogId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-CatalogId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-CatalogId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-InputObject] <Catalog[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-InputObject] <Catalog[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-CatalogName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-CatalogName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

## Detailed Description

The Set-BrokerCatalogMetadata cmdlet creates/updates metadata key-value pairs for Catalog. The Catalog can be specified by InputObject or piping.

## Related topics

### Parameters

**-CatalogId**<Int32>

Specifies the Catalog object whose Metadata is to be created/updated by ID.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Catalog[]>



Specifies the Catalog objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-CatalogName**<String>

Specifies the Catalog object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerCatalog You can pipe the Catalog to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerCatalog

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerCatalog object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerCatalogMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Catalog whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerCatalog | Set-BrokerCatalogMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Catalog in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerCatalogMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Catalog in the site whose name is 'objname'

# Set-BrokerConfigurationSlotMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for ConfigurationSlot

## Syntax

```
Set-BrokerConfigurationSlotMetadata [-ConfigurationSlotId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-ConfigurationSlotId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-ConfigurationSlotId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-InputObject] <ConfigurationSlot[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-InputObject] <ConfigurationSlot[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-ConfigurationSlotName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-ConfigurationSlotName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerConfigurationSlotMetadata cmdlet creates/updates metadata key-value pairs for ConfigurationSlot. The ConfigurationSlot can be specified by InputObject or piping.

## Related topics

### Parameters

**-ConfigurationSlotId**<Int32>

Specifies the ConfigurationSlot object whose Metadata is to be created/updated by ID.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-InputObject**<ConfigurationSlot[]>

Specifies the ConfigurationSlot objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-ConfigurationSlotName**<String>

Specifies the ConfigurationSlot object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerConfigurationSlot You can pipe the ConfigurationSlot to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerConfigurationSlot

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerConfigurationSlot object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerConfigurationSlotMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the ConfigurationSlot whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerConfigurationSlot | Set-BrokerConfigurationSlotMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the ConfigurationSlot in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerConfigurationSlotMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the ConfigurationSlot in the site whose name is 'objname'

# Set-BrokerControllerMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for Controller

## Syntax

```
Set-BrokerControllerMetadata [-ControllerId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerControllerMetadata [-ControllerId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerControllerMetadata [-ControllerId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerControllerMetadata [-InputObject] <Controller[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerControllerMetadata [-InputObject] <Controller[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerControllerMetadata [-ControllerName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerControllerMetadata [-ControllerName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerControllerMetadata cmdlet creates/updates metadata key-value pairs for Controller. The Controller can be specified by InputObject or piping.

## Related topics

### Parameters

**-ControllerId**<Int32>

Specifies the Controller object whose Metadata is to be created/updated by ID.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-InputObject**<Controller[]>



Specifies the Controller objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-ControllerName**<String>

Specifies the Controller object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerController You can pipe the Controller to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerController

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerController object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerControllerMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Controller whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerController | Set-BrokerControllerMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Controller in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerControllerMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Controller in the site whose name is 'objname'

# Set-BrokerDBConnection

Jan 04, 2017

Configures a database connection for the Broker Service.

## Syntax

```
Set-BrokerDBConnection [-DBConnection] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [[-Force]]  
[<CommonParameters>]
```

## Detailed Description

Specifies the database connection string for use by the currently selected Citrix Broker Service instance.

The service records the connection string and attempts to contact the specified database. If the database cannot initially be contacted the service retries the connection at intervals until contact with the database is successfully established.

It is not possible to set a new database connection string for the service if one is already recorded. The connection string must first be set to \$null. This action causes the service to reset and return to its idle state, at which point a new connection string can be set.

When a database connection string is successfully set for the service, a status of OK is returned by the cmdlet. If the database connection string is set to \$null, a DBUnconfigured status is returned.

A syntactically invalid connection string is not recorded.

Only use of Windows authentication within the connection string is supported; a connection string containing SQL authentication credentials is always rejected as invalid.

The current service instance is the one on the local machine, or the one most recently specified using the -AdminAddress parameter of a Broker SDK cmdlet.

## Related topics

[Get-BrokerServiceStatus](#)

[Get-BrokerDBConnection](#)

[Test-BrokerDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be used by the Broker Service. Passing in \$null will clear any existing database connection configured.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**-Force**<SwitchParameter>

If present, allows the local administrator to set the connection string to null when there are problems contacting the database or other services.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Set-BrokerDBConnection cmdlet returns an object describing the status of the Broker Service together with extra diagnostics information. Possible values are:

-- OK:

The Broker Service instance is configured with a valid database and service schema. The service is operational.

-- DBUnconfigured:

No database connection string is set for the Broker Service instance.

-- DBRejectedConnection:

The database rejected the logon attempt from the Broker Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

-- InvalidDBConfigured:

The specified database does not exist, is not visible to the Broker Service instance, or the service's schema within the database is invalid.

-- DBNotFound:

The specified database could not be located with the configured connection string.

-- DBNewerVersionThanService:

The version of the Broker Service currently in use is newer than, and incompatible with, the version of the Broker Service schema on the database. Upgrade the Broker Service to a more recent version.

-- DBOlderVersionThanService:

The version of the Broker Service schema on the database is newer than, and incompatible with, the version of the Broker Service currently in use. Upgrade the database schema to a more recent version.

-- DBVersionChangeInProgress:

A database schema upgrade is in progress.

-- PendingFailure:

Connectivity between the Broker Service instance and the database has been lost. This may be a transitory network error, but may indicate a loss of connectivity that requires administrator intervention.

-- Failed:

Connectivity between the Broker Service instance and the database has been lost for an extended period of time, or has failed due to a configuration problem. The service instance cannot operate while its connection to the database is unavailable.

-- Unknown:

The status of the Broker Service cannot be determined.

Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### InvalidDBConnectionString

The database connection string has an invalid format.

#### DatabaseConnectionDetailsAlreadyConfigured

There was already a database connection configured. After a configuration is set, it can only be set to \$null.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
C:\PS>Set-BrokerDBConnection "Server=dbserver\SQLEXPRESS;Database=XDDB;Trusted_Connection=True"
```

Configures the service instance to use a database called XDDB on an SQL Server Express database running on the machine called dbserver. Integrated Windows authentication is required.

##### ----- EXAMPLE 2 -----

```
C:\PS>Set-BrokerDBConnection -DBConnection $null
```

Resets the service instance's database connection string. The service instance resets and returns to an idle state until a valid new database connection string is specified.

# Set-BrokerDesktopGroup

Jan 04, 2017

Adjusts the settings of a broker desktop group.

## Syntax

```
Set-BrokerDesktopGroup [-InputObject] <DesktopGroup[]> [-PassThru] [-AutomaticPowerOnForAssigned <Boolean>] [-AutomaticPowerOnForAssignedDuringPeak <Boolean>] [-ColorDepth <ColorDepth>] [-DeliveryType <DeliveryType>] [-Description <String>] [-Enabled <Boolean>] [-IconUid <Int32>] [-InMaintenanceMode <Boolean>] [-IsRemotePC <Boolean>] [-MinimumFunctionalLevel <FunctionalLevel>] [-OffPeakBufferSizePercent <Int32>] [-OffPeakDisconnectAction <SessionChangeHostingAction>] [-OffPeakDisconnectTimeout <Int32>] [-OffPeakExtendedDisconnectAction <SessionChangeHostingAction>] [-OffPeakExtendedDisconnectTimeout <Int32>] [-OffPeakLogOffAction <SessionChangeHostingAction>] [-OffPeakLogOffTimeout <Int32>] [-PeakBufferSizePercent <Int32>] [-PeakDisconnectAction <SessionChangeHostingAction>] [-PeakDisconnectTimeout <Int32>] [-PeakExtendedDisconnectAction <SessionChangeHostingAction>] [-PeakExtendedDisconnectTimeout <Int32>] [-PeakLogOffAction <SessionChangeHostingAction>] [-PeakLogOffTimeout <Int32>] [-ProtocolPriority <String[]>] [-PublishedName <String>] [-SecureIcaRequired <Boolean>] [-SettlementPeriodBeforeAutoShutdown <TimeSpan>] [-ShutdownDesktopsAfterUse <Boolean>] [-TimeZone <String>] [-TurnOnAddedMachine <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroup [-Name] <String> [-PassThru] [-AutomaticPowerOnForAssigned <Boolean>] [-AutomaticPowerOnForAssignedDuringPeak <Boolean>] [-ColorDepth <ColorDepth>] [-DeliveryType <DeliveryType>] [-Description <String>] [-Enabled <Boolean>] [-IconUid <Int32>] [-InMaintenanceMode <Boolean>] [-IsRemotePC <Boolean>] [-MinimumFunctionalLevel <FunctionalLevel>] [-OffPeakBufferSizePercent <Int32>] [-OffPeakDisconnectAction <SessionChangeHostingAction>] [-OffPeakDisconnectTimeout <Int32>] [-OffPeakExtendedDisconnectAction <SessionChangeHostingAction>] [-OffPeakExtendedDisconnectTimeout <Int32>] [-OffPeakLogOffAction <SessionChangeHostingAction>] [-OffPeakLogOffTimeout <Int32>] [-PeakBufferSizePercent <Int32>] [-PeakDisconnectAction <SessionChangeHostingAction>] [-PeakDisconnectTimeout <Int32>] [-PeakExtendedDisconnectAction <SessionChangeHostingAction>] [-PeakExtendedDisconnectTimeout <Int32>] [-PeakLogOffAction <SessionChangeHostingAction>] [-PeakLogOffTimeout <Int32>] [-ProtocolPriority <String[]>] [-PublishedName <String>] [-SecureIcaRequired <Boolean>] [-SettlementPeriodBeforeAutoShutdown <TimeSpan>] [-ShutdownDesktopsAfterUse <Boolean>] [-TimeZone <String>] [-TurnOnAddedMachine <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerDesktopGroup cmdlet is used to disable or enable an existing broker desktop group or to alter its settings.

## Related topics

[Get-BrokerDesktopGroup](#)

[New-BrokerDesktopGroup](#)

[Rename-BrokerDesktopGroup](#)

[Remove-BrokerDesktopGroup](#)



## Parameters

### **-InputObject**<DesktopGroup[]>

Specifies the desktop groups to adjust.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

### **-Name**<String>

Specifies the desktop groups to adjust, based on their Name property.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

### **-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

### **-AutomaticPowerOnForAssigned**<Boolean>

Specifies whether assigned desktops in the desktop group should be automatically started at the start of peak time periods. Only relevant for groups whose DesktopKind is Private.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AutomaticPowerOnForAssignedDuringPeak**<Boolean>

Specifies whether assigned desktops in the desktop group should be automatically started throughout peak time periods. Only relevant for groups whose DesktopKind is Private and which have AutomaticPowerOnForAssigned set to true.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ColorDepth**<ColorDepth>

Specifies the color depth that the ICA session should use for desktops in this group. Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DeliveryType**<DeliveryType>

Specifies whether desktops, applications, or both, can be delivered from machines contained within the desktop group. Desktop groups with a DesktopKind of Private cannot be used to deliver both desktops and applications.

When changing the delivery type to desktops only, there must be no remaining desktop-hosted applications associated with the group, or application-specific assignment/entitlement policy rules for the group.

When changing the delivery type to applications only, there must be no remaining client-hosted applications associated with the group, or desktop-specific assignment/entitlement policy rules for the group.

Valid values are DesktopsOnly, AppsOnly, and DesktopsAndApps.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

A description for this desktop group useful for administrators of the site.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Whether the desktop group should be in the enabled state; disabled desktop groups do not appear to users.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconUid**<Int32>

The UID of the broker icon to be displayed to users for their desktop(s) in this desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InMaintenanceMode**<Boolean>

Whether the desktop should be put into maintenance mode; a desktop group in maintenance mode will not allow users to connect or reconnect to their desktops.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsRemotePC**<Boolean>

Supplies a new value for IsRemotePC.

IsRemotePC can only be enabled when:

- o SessionSupport is SingleSession
- o DeliveryType is DesktopsOnly
- o DesktopKind is Private

IsRemotePC can be switched from true to false only if no RemotePC relationship exists between a catalog and this desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MinimumFunctionalLevel**<FunctionalLevel>

The new minimum FunctionalLevel required for machines to work successfully in the desktop group. If this is higher than the FunctionalLevel of any machines already in the desktop group, they will immediately cease to function.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OffPeakBufferSizePercent**<Int32>

The percentage of machines in the desktop group that should be kept available in an idle state outside peak hours.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OffPeakDisconnectAction**<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session disconnecting outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OffPeakDisconnectTimeout**<Int32>

The number of minutes before the configured action should be performed after a user session disconnects outside peak hours.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OffPeakExtendedDisconnectAction**<SessionChangeHostingAction>

The action to be performed after a second configurable period of a user session disconnecting outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OffPeakExtendedDisconnectTimeout**<Int32>

The number of minutes before the second configured action should be performed after a user session disconnects outside peak hours.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OffPeakLogOffAction**<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session ending outside peak hours. Possible values are Nothing, Suspend, or Shutdown.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OffPeakLogOffTimeout**<Int32>

The number of minutes before the configured action should be performed after a user session ends outside peak hours.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakBufferSizePercent**<Int32>

The percentage of machines in the desktop group that should be kept available in an idle state in peak hours.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakDisconnectAction**<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session disconnecting in peak hours. Possible values are Nothing, Suspend, or Shutdown.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakDisconnectTimeout**<Int32>

The number of minutes before the configured action should be performed after a user session disconnects in peak hours.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakExtendedDisconnectAction**<SessionChangeHostingAction>

The action to be performed after a second configurable period of a user session disconnecting in peak hours. Possible values are Nothing, Suspend, or Shutdown.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakExtendedDisconnectTimeout**<Int32>

The number of minutes before the second configured action should be performed after a user session disconnects in peak hours.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakLogOffAction**<SessionChangeHostingAction>

The action to be performed after a configurable period of a user session ending in peak hours. Possible values are Nothing, Suspend, or Shutdown.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakLogOffTimeout**<Int32>

The number of minutes before the configured action should be performed after a user session ends in peak hours.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ProtocolPriority**<String[]>

A list of protocol names in the order in which they should be attempted for use during connection.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName**<String>

The name that will be displayed to users for their desktop(s) in this desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaRequired**<Boolean>

Whether HDX connections to desktops in the new desktop group require the use of a secure protocol.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SettlementPeriodBeforeAutoShutdown**<TimeSpan>



Time after a session ends during which automatic shutdown requests (for example, shutdown after use, idle pool management) are deferred. Any outstanding shutdown request takes effect after the settlement period expires. This is typically used to configure time to allow logoff scripts to complete.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ShutdownDesktopsAfterUse**<Boolean>

Whether desktops in this desktop group should be automatically shut down when each user session completes (only relevant to power-managed desktops).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TimeZone**<String>

The time zone in which this desktop group's machines reside.

The time zone must be specified for any of the group's automatic power management settings to take effect. Automatic power management operations include pool management (power time schemes), reboot schedules, session disconnect and logoff actions, and powering on assigned machines etc.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TurnOnAddedMachine**<Boolean>

This flag specifies whether the Broker Service should attempt to power on machines when they are added to the desktop group.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.DesktopGroup You can pipe the desktop groups to be adjusted to Set-BrokerDesktopGroup.

**Return Values**

None or Citrix.Broker.Admin.SDK.DesktopGroup

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.DesktopGroup object.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerDesktopGroup EMEA* -InMaintenanceMode $true -PassThru
```

Sets all desktop groups with names starting "EMEA" into maintenance mode, returning the set of desktop groups.

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerDesktopGroup -InMaintenanceMode $true | Set-BrokerDesktopGroup -Enabled $false
```

Disable all desktop groups that are in maintenance mode.

# Set-BrokerDesktopGroupMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for DesktopGroup

## Syntax

```
Set-BrokerDesktopGroupMetadata [-DesktopGroupId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroupMetadata [-DesktopGroupId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroupMetadata [-DesktopGroupId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroupMetadata [-InputObject] <DesktopGroup[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroupMetadata [-InputObject] <DesktopGroup[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroupMetadata [-DesktopGroupName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroupMetadata [-DesktopGroupName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerDesktopGroupMetadata cmdlet creates/updates metadata key-value pairs for DesktopGroup. The DesktopGroup can be specified by InputObject or piping.

## Related topics

### Parameters

**-DesktopGroupId**<Int32>

Specifies the DesktopGroup object whose Metadata is to be created/updated by ID.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<DesktopGroup[]>

Specifies the DesktopGroup objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-DesktopGroupName**<String>

Specifies the DesktopGroup object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerDesktopGroup You can pipe the DesktopGroup to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerDesktopGroup

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerDesktopGroup object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerDesktopGroupMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the DesktopGroup whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerDesktopGroup | Set-BrokerDesktopGroupMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the DesktopGroup in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerDesktopGroupMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the DesktopGroup in the site whose name is 'objname'

# Set-BrokerEntitlementPolicyRule

Jan 04, 2017

Modifies an existing desktop rule in the site's entitlement policy.

## Syntax

```
Set-BrokerEntitlementPolicyRule [-InputObject] <EntitlementPolicyRule[]> [-PassThru] [-AddExcludedUsers <User[]>] [-AddIncludedUsers <User[]>] [-ColorDepth <ColorDepth>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IconUid <Int32>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-PublishedName <String>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedUsers <User[]>] [-SecureIcaRequired <Boolean>] [-SessionReconnection <SessionReconnection>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRule [-Name] <String> [-PassThru] [-AddExcludedUsers <User[]>] [-AddIncludedUsers <User[]>] [-ColorDepth <ColorDepth>] [-Description <String>] [-Enabled <Boolean>] [-ExcludedUserFilterEnabled <Boolean>] [-ExcludedUsers <User[]>] [-IconUid <Int32>] [-IncludedUserFilterEnabled <Boolean>] [-IncludedUsers <User[]>] [-PublishedName <String>] [-RemoveExcludedUsers <User[]>] [-RemoveIncludedUsers <User[]>] [-SecureIcaRequired <Boolean>] [-SessionReconnection <SessionReconnection>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerEntitlementPolicyRule cmdlet modifies an existing desktop rule in the site's entitlement policy.

A desktop rule in the entitlement policy defines the users who are allowed per-session access to a machine from the rule's associated desktop group to run a full desktop session.

Changing a rule does not affect existing sessions launched using the rule, but if the change removes an entitlement to a machine that was previously granted, users may be unable to reconnect to a disconnected session on that machine.

## Related topics

[New-BrokerEntitlementPolicyRule](#)

[Get-BrokerEntitlementPolicyRule](#)

[Rename-BrokerEntitlementPolicyRule](#)

[Remove-BrokerEntitlementPolicyRule](#)

## Parameters

**-InputObject**<EntitlementPolicyRule[]>

The desktop rule in the entitlement policy to be modified.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |



|                        |                |
|------------------------|----------------|
| Accept Pipeline Input? | true (ByValue) |
|------------------------|----------------|

**-Name**<String>

The name of the desktop rule in the entitlement policy to be modified.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-AddExcludedUsers**<User[]>

Adds the specified users to the excluded users filter of the rule, that is, the users and groups who are explicitly denied an entitlement to a desktop session from this rule.

See the ExcludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AddIncludedUsers**<User[]>

Adds the specified users to the included users filter of the rule, that is, the users and groups who are granted an entitlement to a desktop session by the rule.

See the IncludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ColorDepth**<ColorDepth>

Changes the color depth of any desktop sessions launched by a user from this entitlement. Existing sessions are not affected.

Valid values are \$null, FourBit, EightBit, SixteenBit, and TwentyFourBit.

A null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Changes the description of the desktop rule. The text may be visible to the end user, for example, as a tooltip associated with the desktop entitlement.

A null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Enables or disables the desktop rule. A disabled rule is ignored when evaluating the site's entitlement policy.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUserFilterEnabled**<Boolean>

Enables or disables the excluded users filter. If the filter is disabled then any user entries in the filter are ignored when entitlement policy rules are evaluated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ExcludedUsers**<User[]>

Changes the excluded users filter of the desktop rule, that is, the users and groups who are explicitly denied an entitlement to a desktop session from this rule.

This can be used to exclude users or groups who would otherwise gain access by groups specified in the included users filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconUid**<Int32>

Changes the icon (identified by its unique ID) for the published desktop entitlement as seen by the user.

A null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUserFilterEnabled**<Boolean>

Enables or disables the included users filter. If the filter is disabled then any user who satisfies the requirements of the access policy is implicitly granted an entitlement to a desktop session by the rule.

Users who would be implicitly granted access when the filter is disabled can still be explicitly denied access using the excluded users filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IncludedUsers**<User[]>

Changes the included users filter of the desktop rule, that is, the users and groups who are granted an entitlement to a desktop session by the rule.

If a user appears explicitly in the excluded users filter of the rule or is a member of a group that appears in the excluded users filter, no entitlement is granted whether or not the user appears in the included users filter.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName**<String>

Changes the name of the published desktop entitlement as seen by the user.

A null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-RemoveExcludedUsers**<User[]>

Removes the specified users from the excluded users filter of the rule, that is, the users and groups who are explicitly denied an entitlement to a desktop session from this rule.

See the ExcludedUsers parameter for more information.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-RemoveIncludedUsers**<User[]>

Removes the specified users from the included users filter of the desktop rule, that is, the users and groups who are granted an entitlement to a desktop session by the rule.

See the IncludedUsers parameter for more information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaRequired**<Boolean>

Changes whether the desktop rule requires the SecureICA protocol for desktop sessions launched using the entitlement.

A null value indicates that the equivalent setting from the rule's desktop group is used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SessionReconnection**<SessionReconnection>

Defines reconnection (roaming) behavior for sessions launched using this rule. Session reconnection control is an experimental and unsupported feature.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.EntitlementPolicyRule The desktop rule in the entitlement policy to be modified.

**Return Values**

None or Citrix.Broker.Admin.SDK.EntitlementPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.EntitlementPolicyRule object.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Set-BrokerEntitlementPolicyRule 'Temp Workers' -AddIncludedUsers office\contractors

Adds the user group OFFICE\contractors to the Temp Workers desktop rule of the entitlement policy. This grants all members of that group an entitlement to a desktop session in the rule's associated desktop group. The session properties of the desktops obtained using the rule are determined by the rule's other properties.

----- **EXAMPLE 2** -----

C:\PS> Set-BrokerEntitlementPolicyRule 'Temp Workers' -Enabled \$false

Disables the Temp Workers desktop rule in the entitlement policy. This prevents further desktop sessions being launched using this rule until it is re-enabled. However, access to existing desktop sessions is not affected.

# Set-BrokerEntitlementPolicyRuleMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for EntitlementPolicyRule

## Syntax

```
Set-BrokerEntitlementPolicyRuleMetadata [-EntitlementPolicyRuleId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRuleMetadata [-EntitlementPolicyRuleId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRuleMetadata [-EntitlementPolicyRuleId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRuleMetadata [-InputObject] <EntitlementPolicyRule[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRuleMetadata [-InputObject] <EntitlementPolicyRule[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRuleMetadata [-EntitlementPolicyRuleName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRuleMetadata [-EntitlementPolicyRuleName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerEntitlementPolicyRuleMetadata cmdlet creates/updates metadata key-value pairs for EntitlementPolicyRule. The EntitlementPolicyRule can be specified by InputObject or piping.

## Related topics

### Parameters

**-EntitlementPolicyRuleId**<Int32>

Specifies the EntitlementPolicyRule object whose Metadata is to be created/updated by ID.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-InputObject**<EntitlementPolicyRule[]>

Specifies the EntitlementPolicyRule objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-EntitlementPolicyRuleName**<String>

Specifies the EntitlementPolicyRule object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|



|                        |                |
|------------------------|----------------|
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.BrokerEntitlementPolicyRule You can pipe the EntitlementPolicyRule to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerEntitlementPolicyRule

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerEntitlementPolicyRule object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerEntitlementPolicyRuleMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the EntitlementPolicyRule whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerEntitlementPolicyRule | Set-BrokerEntitlementPolicyRuleMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the EntitlementPolicyRule in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerEntitlementPolicyRuleMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the EntitlementPolicyRule in the site whose name is 'objname'

# Set-BrokerHostingPowerAction

Jan 04, 2017

Changes the priority of one or more pending power actions.

## Syntax

```
Set-BrokerHostingPowerAction [-InputObject] <HostingPowerAction[]> [-PassThru] [-ActualPriority <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHostingPowerAction [-MachineName] <String> [-PassThru] [-ActualPriority <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerHostingPowerAction cmdlet modifies an existing power action in the site's power action queue. The only property of power actions you can change, is the current priority of the action.

For a detailed description of the queuing mechanism, see 'help about\_Broker\_PowerManagement'.

## Related topics

[Get-BrokerHostingPowerAction](#)

[New-BrokerHostingPowerAction](#)

[Remove-BrokerHostingPowerAction](#)

## Parameters

**-InputObject** <HostingPowerAction[]>

The power action whose priority is to be changed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-MachineName** <String>

Changes the priority of actions that are for machines whose name (of the form domain\machine) matches the specified string.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-ActualPriority**<Int32>

Specifies a new priority value for the action in the queue.

This priority is the current action priority; the 'base' or original priority for actions cannot be altered. Numerically lower priority values indicate more important actions that are processed in preference to actions with numerically higher priority settings.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host

name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.HostingPowerAction The power action whose priority is to be changed.

### Return Values

None or Citrix.Broker.Admin.SDK.HostingPowerAction

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.HostingPowerAction object.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerHostingPowerAction -MachineName 'XD_VDA1' -ActualPriority 25
```

Sets the current priority of actions for the machine called 'XD\_VDA1' to 25. Numerically lower priority values indicate more important actions that will be processed in preference to actions with numerically higher priority settings.

# Set-BrokerHostingPowerActionMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for HostingPowerAction

## Syntax

```
Set-BrokerHostingPowerActionMetadata [-HostingPowerActionId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHostingPowerActionMetadata [-HostingPowerActionId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHostingPowerActionMetadata [-HostingPowerActionId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHostingPowerActionMetadata [-InputObject] <HostingPowerAction[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHostingPowerActionMetadata [-InputObject] <HostingPowerAction[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHostingPowerActionMetadata [-HostingPowerActionName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHostingPowerActionMetadata [-HostingPowerActionName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerHostingPowerActionMetadata cmdlet creates/updates metadata key-value pairs for HostingPowerAction. The HostingPowerAction can be specified by InputObject or piping.

## Related topics

### Parameters

**-HostingPowerActionId**<Int64>

Specifies the HostingPowerAction object whose Metadata is to be created/updated by ID.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<HostingPowerAction[]>

Specifies the HostingPowerAction objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-HostingPowerActionName**<String>

Specifies the HostingPowerAction object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type



Citrix.Broker.Admin.SDK.BrokerHostingPowerAction You can pipe the HostingPowerAction to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerHostingPowerAction

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerHostingPowerAction object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerHostingPowerActionMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the HostingPowerAction whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerHostingPowerAction | Set-BrokerHostingPowerActionMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the HostingPowerAction in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerHostingPowerActionMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the HostingPowerAction in the site whose name is 'objname'

# Set-BrokerHypervisorAlertMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for HypervisorAlert

## Syntax

```
Set-BrokerHypervisorAlertMetadata [-HypervisorAlertId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorAlertMetadata [-HypervisorAlertId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorAlertMetadata [-HypervisorAlertId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorAlertMetadata [-InputObject] <HypervisorAlert[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorAlertMetadata [-InputObject] <HypervisorAlert[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerHypervisorAlertMetadata cmdlet creates/updates metadata key-value pairs for HypervisorAlert. The HypervisorAlert can be specified by InputObject or piping.

## Related topics

### Parameters

**-HypervisorAlertId**<Int64>

Specifies the HypervisorAlert object whose Metadata is to be created/updated by ID.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-InputObject**<HypervisorAlert[]>

Specifies the HypervisorAlert objects whose Metadata is to be created/updated.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                |
|------------------------|----------------|
| Accept Pipeline Input? | true (ByValue) |
|------------------------|----------------|

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | False |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.BrokerHypervisorAlert You can pipe the HypervisorAlert to hold the new or updated metadata.

**Return Values**

None or Citrix.Broker.Admin.SDK.BrokerHypervisorAlert

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerHypervisorAlert object.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerHypervisorAlertMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
This command creates/updates the Metadata "MyMetadataName" key-value pair for the HypervisorAlert whose instance is pointed by $obj-Uid
```

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerHypervisorAlert | Set-BrokerHypervisorAlertMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the HypervisorAlert in the site

# Set-BrokerHypervisorConnection

Jan 04, 2017

Sets the properties of a hypervisor connection.

## Syntax

```
Set-BrokerHypervisorConnection [-InputObject] <HypervisorConnection[]> [-PassThru] [-PreferredController <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnection [-Name] <String> [-PassThru] [-PreferredController <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerHypervisorConnection cmdlet sets the properties on a hypervisor connection.

## Related topics

[Get-BrokerHypervisorConnection](#)

[New-BrokerHypervisorConnection](#)

[Remove-BrokerHypervisorConnection](#)

## Parameters

**-InputObject** <HypervisorConnection[]>

Specifies the hypervisor connection object to adjust.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name** <String>

Specifies the name of the hypervisor connection object to adjust.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-PreferredController**<String>

Supplies the new value of the PreferredController property.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|               |                                                                                        |
|---------------|----------------------------------------------------------------------------------------|
| Required?     | false                                                                                  |
| Default Value | localhost. Once a value is provided by any cmdlet, this value will become the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Input Type

Citrix.Broker.Admin.SDK.HypervisorConnection You can pipe the hypervisor connection to be modified to Set-BrokerHypervisorConnection.

## Return Values

None or Citrix.Broker.Admin.SDK.HypervisorConnection

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.HypervisorConnection object.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS> $hvConn = Get-BrokerHypervisorConnection -PreferredController "oldController" -Name "Xen Server Connection"
c:\PS> Set-BrokerHypervisorConnection -InputObject $hvConn -PreferredController "newController"
Updates the preferred controller for a hypervisor connection.
```



# Set-BrokerHypervisorConnectionMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for HypervisorConnection

## Syntax

```
Set-BrokerHypervisorConnectionMetadata [-HypervisorConnectionId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnectionMetadata [-HypervisorConnectionId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnectionMetadata [-HypervisorConnectionId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerHypervisorConnectionMetadata cmdlet creates/updates metadata key-value pairs for HypervisorConnection. The HypervisorConnection can be specified by InputObject or piping.

## Related topics

### Parameters

**-HypervisorConnectionId**<Int32>

Specifies the HypervisorConnection object whose Metadata is to be created/updated by ID.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<HypervisorConnection[]>

Specifies the HypervisorConnection objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-HypervisorConnectionName**<String>

Specifies the HypervisorConnection object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.BrokerHypervisorConnection You can pipe the HypervisorConnection to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerHypervisorConnection

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerHypervisorConnection object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerHypervisorConnectionMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the HypervisorConnection whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerHypervisorConnection | Set-BrokerHypervisorConnectionMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the HypervisorConnection in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerHypervisorConnectionMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the HypervisorConnection in the site whose name is 'objname'

# Set-BrokerIconMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for Icon

## Syntax

```
Set-BrokerIconMetadata [-IconId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [  
<CommonParameters>]
```

```
Set-BrokerIconMetadata [-IconId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress  
<String>] [  
<CommonParameters>]
```

```
Set-BrokerIconMetadata [-IconId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress  
<String>] [  
<CommonParameters>]
```

```
Set-BrokerIconMetadata [-InputObject] <Icon[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-  
AdminAddress <String>] [  
<CommonParameters>]
```

```
Set-BrokerIconMetadata [-InputObject] <Icon[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [  
<CommonParameters>]
```

## Detailed Description

The Set-BrokerIconMetadata cmdlet creates/updates metadata key-value pairs for Icon. The Icon can be specified by InputObject or piping.

## Related topics

### Parameters

**-IconId**<Int32>

Specifies the Icon object whose Metadata is to be created/updated by ID.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Icon[]>

Specifies the Icon objects whose Metadata is to be created/updated.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                |
|------------------------|----------------|
| Accept Pipeline Input? | true (ByValue) |
|------------------------|----------------|

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | False |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.BrokerIcon You can pipe the Icon to hold the new or updated metadata.

**Return Values**

None or Citrix.Broker.Admin.SDK.BrokerIcon

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerIcon object.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerIconMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Icon whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerIcon | Set-BrokerIconMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Icon in the site



# Set-BrokerLeaseMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for Lease

## Syntax

```
Set-BrokerLeaseMetadata [-LeaseId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [  
<CommonParameters>]
```

```
Set-BrokerLeaseMetadata [-LeaseId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-  
AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerLeaseMetadata [-LeaseId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-  
AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerLeaseMetadata [-InputObject] <Lease[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-BrokerLeaseMetadata [-InputObject] <Lease[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-  
AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerLeaseMetadata [-LeaseName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-BrokerLeaseMetadata [-LeaseName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-  
AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerLeaseMetadata cmdlet creates/updates metadata key-value pairs for Lease. The Lease can be specified by InputObject or piping.

## Related topics

### Parameters

**-LeaseId**<Int64>

Specifies the Lease object whose Metadata is to be created/updated by ID.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Lease[]>

Specifies the Lease objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LeaseName**<String>

Specifies the Lease object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerLease You can pipe the Lease to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerLease

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerLease object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerLeaseMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Lease whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerLease | Set-BrokerLeaseMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Lease in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerLeaseMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Lease in the site whose name is 'objname'

# Set-BrokerMachine

Jan 04, 2017

Sets properties on a machine.

## Syntax

```
Set-BrokerMachine [-InputObject] <Machine[]> [-PassThru] [-AssignedClientName <String>] [-AssignedIPAddress <String>] [-HostedMachineId <String>] [-HypervisorConnectionUid <Int32>] [-InMaintenanceMode <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachine [-MachineName] <String> [-PassThru] [-AssignedClientName <String>] [-AssignedIPAddress <String>] [-HostedMachineId <String>] [-HypervisorConnectionUid <Int32>] [-InMaintenanceMode <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerMachine cmdlet sets properties on a machine or set of machines. You can specify a single machine by name or multiple machine instances can be passed to the command by piping or using the -InputObject parameter.

## Related topics

[Get-BrokerMachine](#)

[New-BrokerMachine](#)

## Parameters

**-InputObject** <Machine[]>

The machine instances whose properties you want to set.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-MachineName** <String>

The machine whose properties you want to set.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-AssignedClientName**<String>

Changes the client name assignment of the machine. Set this to \$null to remove the assignment. You can assign machines to multiple users, a single client IP address, or a single client name, but only to one of these categories at a time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedIPAddress**<String>

Changes the client IP address assignment of the machine. Set this to \$null to remove the assignment. You can assign machines to multiple users, a single client IP address, or a single client name, but only to one of these categories at a time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HostedMachineId**<String>

The unique ID by which the hypervisor recognizes the machine. This may only be set for VMs which are not provisioned by MCS.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-HypervisorConnectionUid<Int32>**

The hypervisor connection that runs the machine. This may only be set for VMs which are not provisioned by MCS.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InMaintenanceMode<Boolean>**

Sets whether the machine is in maintenance mode or not. A machine in maintenance mode is not available for new sessions, and for managed machines all automatic power management is disabled.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

Citrix.Broker.Admin.SDK.Machine You can pipe in the machines whose properties you want to set.

## Return Values

None or Citrix.Broker.Admin.SDK.Machine

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.Machine object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerMachine -MachineName 'MyDomain\MyMachine' -HypervisorConnectionUid 3
```

This example specifies a machine by name and sets its hypervisor connection.

### ----- EXAMPLE 2 -----

```
C:\PS> $machines = Get-BrokerMachine -MachineName 'MyDomain*'
```

```
C:\PS> Set-BrokerMachine -InputObject $machines -HypervisorConnectionUid 3
```

This example finds all machines in domain MyDomain and sets their hypervisor connections.

### ----- EXAMPLE 3 -----

```
C:\PS> Get-BrokerMachine -MachineName 'MyDomain*' | Set-BrokerMachine -HypervisorConnectionUid 3
```

This example also finds all machines in domain MyDomain and sets their hypervisor connections.



# Set-BrokerMachineCatalog

Jan 04, 2017

Moves one or more machines into a different catalog.

## Syntax

```
Set-BrokerMachineCatalog [-InputObject] <Machine[]> [-CatalogUid] <Int32> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerMachineCatalog cmdlet is used to move machines into a different catalog. The following properties of the destination catalog must exactly match those of the machine's current catalog otherwise the command fails:

- o AllocationType
- o ProvisioningType
- o PersistUserChanges
- o SessionSupport
- o IsRemotePC
- o MinimumFunctionalLevel
- o PhysicalMachines

Changing a machine's catalog does not change the machine's desktop group membership. There is no effect on user sessions present on a machine if its catalog is changed.

## Related topics

[Set-BrokerMachine](#)

[New-BrokerCatalog](#)

## Parameters

**-InputObject**<Machine[]>

The machine instances that are being moved into a different catalog.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-CatalogUid**<Int32>

The unique identifier of the catalog into which the machines are being moved.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.Machine You can pipe in the machines that are to be moved into a new catalog.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> $catalog = Get-BrokerCatalog MarketingMachines
C:\PS> $machines = Get-BrokerMachine -MachineName 'Marketing\*'
C:\PS> Set-BrokerMachineCatalog $machines -CatalogUid $cat.Uid
```

This example finds all machines in domain Marketing and moves them into a catalog called MarketingMachines.

# Set-BrokerMachineCommandMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for MachineCommand

## Syntax

```
Set-BrokerMachineCommandMetadata [-MachineCommandId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineCommandMetadata [-MachineCommandId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineCommandMetadata [-MachineCommandId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineCommandMetadata [-InputObject] <MachineCommand[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineCommandMetadata [-InputObject] <MachineCommand[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerMachineCommandMetadata cmdlet creates/updates metadata key-value pairs for MachineCommand. The MachineCommand can be specified by InputObject or piping.

## Related topics

### Parameters

**-MachineCommandId**<Int64>

Specifies the MachineCommand object whose Metadata is to be created/updated by ID.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<MachineCommand[]>

Specifies the MachineCommand objects whose Metadata is to be created/updated.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                |
|------------------------|----------------|
| Accept Pipeline Input? | true (ByValue) |
|------------------------|----------------|

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | False |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.BrokerMachineCommand You can pipe the MachineCommand to hold the new or updated metadata.

**Return Values**

None or Citrix.Broker.Admin.SDK.BrokerMachineCommand

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerMachineCommand object.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerMachineCommandMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
This command creates/updates the Metadata "MyMetadataName" key-value pair for the MachineCommand whose instance is pointed by $obj-Uid
```

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerMachineCommand | Set-BrokerMachineCommandMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the MachineCommand in the site

# Set-BrokerMachineConfiguration

Jan 04, 2017

Sets the properties of a machine configuration.

## Syntax

```
Set-BrokerMachineConfiguration [-InputObject] <MachineConfiguration[]> [-PassThru] [-Description <String>] [-Policy <Byte[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfiguration [-Name] <String> [-PassThru] [-Description <String>] [-Policy <Byte[]>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Sets the properties of a machine configuration. The encoded settings data must only contain settings that match the SettingsGroup of the associated configuration slot. Use the SDK snap-in that matches the SettingsGroup of the associated configuration slot to generate new encoded settings data or modify existing settings values.

## Related topics

[New-BrokerMachineConfiguration](#)

[Get-BrokerMachineConfiguration](#)

[Rename-BrokerMachineConfiguration](#)

[Remove-BrokerMachineConfiguration](#)

[Add-BrokerMachineConfiguration](#)

## Parameters

**-InputObject**<MachineConfiguration[]>

Machine configuration to modify.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Name of machine configuration to modify.

|                        |      |
|------------------------|------|
| Required?              | true |
| Default Value          |      |
| Accept Pipeline Input? |      |



|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-Description**<String>

New description for the machine configuration.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Policy**<Byte[]>

New binary array of encoded settings data.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.MachineConfiguration Machine configuration to modify.

Return Values

None or Citrix.Broker.Admin.SDK.MachineConfiguration

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.MachineConfiguration object.

Examples

----- **EXAMPLE 1** -----

Set-BrokerMachineConfiguration -Name "UPM\Finance Department" -Policy \$newPolicy  
 Use the encoded settings binary data in \$newPolicy to update the machine configuration.

----- **EXAMPLE 2** -----

Get-BrokerMachineConfiguration -Name "UPM\*" | Set-BrokerMachineConfiguration -Description "User Profile Management"  
 Assign the description "User Profile Management" to every machine configuration associated with the UPM slot.

# Set-BrokerMachineConfigurationMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for MachineConfiguration

## Syntax

```
Set-BrokerMachineConfigurationMetadata [-MachineConfigurationId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-MachineConfigurationId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-MachineConfigurationId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-InputObject] <MachineConfiguration[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-InputObject] <MachineConfiguration[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-MachineConfigurationName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-MachineConfigurationName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerMachineConfigurationMetadata cmdlet creates/updates metadata key-value pairs for MachineConfiguration. The MachineConfiguration can be specified by InputObject or piping.

## Related topics

### Parameters

**-MachineConfigurationId**<Int32>

Specifies the MachineConfiguration object whose Metadata is to be created/updated by ID.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-InputObject**<MachineConfiguration[]>

Specifies the MachineConfiguration objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-MachineConfigurationName**<String>

Specifies the MachineConfiguration object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                |
|------------------------|----------------|
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.BrokerMachineConfiguration You can pipe the MachineConfiguration to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerMachineConfiguration

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerMachineConfiguration object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerMachineConfigurationMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the MachineConfiguration whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerMachineConfiguration | Set-BrokerMachineConfigurationMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the MachineConfiguration in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerMachineConfigurationMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the MachineConfiguration in the site whose name is 'objname'

# Set-BrokerMachineMaintenanceMode

Jan 04, 2017

Sets whether the specified machine(s) are in maintenance mode.

## Syntax

```
Set-BrokerMachineMaintenanceMode [-InputObject] <Machine[]> [-MaintenanceMode] <Boolean> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The cmdlet can be used to set whether a machine is in maintenance mode or not. A machine in maintenance mode is not available for new sessions, and for managed machines all automatic power management is disabled.

There are times when it is necessary to disable desktops. You can do this by setting the `InMaintenanceMode` property of a desktop to `$true`. This puts it into maintenance mode. The broker excludes single-session desktops in maintenance mode from brokering decisions and does not start new sessions on them. Existing sessions are unaffected. For multi-session desktops in maintenance mode, reconnections to existing sessions are allowed, but no new sessions are created on the machine.

Desktops in maintenance mode are also excluded from automatic power management, although explicit power actions are still performed.

This cmdlet is equivalent to using the `Set-BrokerMachine` cmdlet to set the value of only the `InMaintenanceMode` property.

## Related topics

[Set-BrokerMachine](#)

## Parameters

**-InputObject** <Machine[]>

The machine instances whose `InMaintenanceMode` property you want to set.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-MaintenanceMode** <Boolean>

Sets whether the machine is in maintenance mode or not. A machine in maintenance mode is not available for new sessions, and for managed machines all automatic power management is disabled.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                |
|------------------------|----------------|
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.Machine You can pipe in the machines whose properties you want to set.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> $machines = Get-BrokerMachine -MachineName 'MyDomain\*'
C:\PS> Set-BrokerMachineMaintenanceMode -InputObject $machines $false
```

```
C:\PS> Set-BrokerMachineMaintenanceMode -InputObject $machines $false
```

This example finds all machines in domain MyDomain and removes them from maintenance mode by setting their InMaintenanceMode property to false.



# Set-BrokerMachineMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for Machine

## Syntax

```
Set-BrokerMachineMetadata [-MachineId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-MachineId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-MachineId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-InputObject] <Machine[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-InputObject] <Machine[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-MachineName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-MachineName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerMachineMetadata cmdlet creates/updates metadata key-value pairs for Machine. The Machine can be specified by InputObject or piping.

## Related topics

### Parameters

**-MachineId**<Int32>

Specifies the Machine object whose Metadata is to be created/updated by ID.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-InputObject**<Machine[]>

Specifies the Machine objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-MachineName**<String>

Specifies the Machine object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerMachine You can pipe the Machine to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerMachine

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerMachine object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerMachineMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Machine whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerMachine | Set-BrokerMachineMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Machine in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerMachineMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Machine in the site whose name is 'objname'

# Set-BrokerPowerTimeScheme

Jan 04, 2017

Modifies an existing power time scheme.

## Syntax

```
Set-BrokerPowerTimeScheme [-InputObject] <PowerTimeScheme[]> [-PassThru] [-DaysOfWeek <TimeSchemeDays>] [-DisplayName <String>] [-PeakHours <Boolean[]>] [-PoolSize <Int32[]>] [-PoolUsingPercentage <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeScheme [-Name] <String> [-PassThru] [-DaysOfWeek <TimeSchemeDays>] [-DisplayName <String>] [-PeakHours <Boolean[]>] [-PoolSize <Int32[]>] [-PoolUsingPercentage <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerPowerTimeScheme cmdlet modifies an existing time scheme.

Each power time scheme is associated with a particular desktop group, and covers one or more days of the week, defining which hours of those days are considered peak times and which are off-peak times. In addition, the time scheme defines a pool size value for each hour of the day for the days of the week covered by the time scheme. No one desktop group can be associated with two or more time schemes that cover the same day of the week.

For more information about the power policy mechanism and pool size management, see 'help about\_Broker\_PowerManagement'.

## Related topics

[Get-BrokerPowerTimeScheme](#)

[Rename-BrokerPowerTimeScheme](#)

[New-BrokerPowerTimeScheme](#)

[Remove-BrokerPowerTimeScheme](#)

## Parameters

**-InputObject** <PowerTimeScheme[]>

The power time scheme to be changed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name** <String>

Identifies the power time scheme to be changed by name.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-DaysOfWeek**<TimeSchemeDays>

Changes the pattern of days of the week that the time scheme applies to. The pattern of days is specified as a single value or a list of values, where each value refers to either a single day or defined group of days.

Valid values are: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Weekend and Weekdays.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DisplayName**<String>

Changes the name that is used by the DesktopStudio console when showing the time scheme.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PeakHours**<Boolean[]>

Changes the pattern of hours considered 'peak' vs 'off-peak' for the days covered by the time scheme. A 24-entry array of boolean truth values is expected, where the zeroth entry of the array relates to the time period between midnight and 0:59, the first relates to 1am to 1:59 and so on, with the last array element relating to 11 PM to 11:59. If the flag value is \$true it means the associated hour of the day is considered a peak time; if \$false it means that it is considered off-peak.

If fewer than 24 values are supplied, the final missing values are assumed to be 'false', and if more than 24 values are supplied, only the first 24 are used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PoolSize**<Int32[]>

Changes the requested pool size of running machines at the various hours of the day for the days covered by the time scheme. A 24-entry array of integer values is expected, where the zeroth entry of the array relates to the time period between midnight and 0:59, the first relates to 1am to 1:59 and so on, with the last array element relating to 11 PM to 11:59. The pool size array entry values are either absolute numbers of machines that should be running or are a percentage of the machines in the desktop group that should be running during the associated hour of the day. A value of -1 in the array signifies that no management of the number of running machines should be attempted during the associated hour of the day.

If fewer than 24 values are supplied, the final missing values are assumed to be -1, and if more than 24 values are supplied, only the first 24 are used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PoolUsingPercentage**<Boolean>

Changes whether pool size values from the 'PoolSize' array are evaluated as absolute numbers of running machines or as a percentage of machines in the desktop group that are to be maintained as running.

A value of \$true indicates that the pool size array values are percentages of total machines in the desktop group and a value of \$false indicates that the pool size array values are absolute numbers of machines to maintain as running.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.PowerTimeScheme The power time scheme to be changed.

**Return Values**

None or Citrix.Broker.Admin.SDK.PowerTimeScheme

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.PowerTimeScheme object.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerPowerTimeScheme -Name 'Development Weekdays' -PoolSize ( 0..23 | %{ if ($_ -lt 8 -or $_ -gt 19) { 5 } else { 20 } } )
```

Sets the pool size for the power time scheme named 'Development Weekdays' to be 20 for the time between 8am to 7:59pm, and 5 for other times.



# Set-BrokerPowerTimeSchemeMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for PowerTimeScheme

## Syntax

```
Set-BrokerPowerTimeSchemeMetadata [-PowerTimeSchemeId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-PowerTimeSchemeId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-PowerTimeSchemeId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-InputObject] <PowerTimeScheme[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-InputObject] <PowerTimeScheme[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-PowerTimeSchemeName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-PowerTimeSchemeName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerPowerTimeSchemeMetadata cmdlet creates/updates metadata key-value pairs for PowerTimeScheme. The PowerTimeScheme can be specified by InputObject or piping.

## Related topics

### Parameters

**-PowerTimeSchemeId**<Int32>

Specifies the PowerTimeScheme object whose Metadata is to be created/updated by ID.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-InputObject**<PowerTimeScheme[]>

Specifies the PowerTimeScheme objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PowerTimeSchemeName**<String>

Specifies the PowerTimeScheme object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerPowerTimeScheme You can pipe the PowerTimeScheme to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerPowerTimeScheme

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerPowerTimeScheme object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerPowerTimeSchemeMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the PowerTimeScheme whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerPowerTimeScheme | Set-BrokerPowerTimeSchemeMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the PowerTimeScheme in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerPowerTimeSchemeMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the PowerTimeScheme in the site whose name is 'objname'

# Set-BrokerPrivateDesktop

Jan 04, 2017

Change the settings of a private desktop.

## Syntax

```
Set-BrokerPrivateDesktop [-InputObject] <PrivateDesktop[]> [-PassThru] [-AssignedClientName <String>] [-AssignedIPAddress <String>] [-ColorDepth <ColorDepth>] [-Description <String>] [-IconUid <Int32>] [-InMaintenanceMode <Boolean>] [-PublishedName <String>] [-SecureIcaRequired <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPrivateDesktop [-MachineName] <String> [-PassThru] [-AssignedClientName <String>] [-AssignedIPAddress <String>] [-ColorDepth <ColorDepth>] [-Description <String>] [-IconUid <Int32>] [-InMaintenanceMode <Boolean>] [-PublishedName <String>] [-SecureIcaRequired <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Private desktops are automatically created when a machine is added to a desktop group with a DesktopKind of 'Private', and these inherit default properties. Use Set-BrokerPrivateDesktop to change the configuration settings of an existing private desktop.

To specify private desktops, you can choose whether to update by machine name, or by passing a PrivateDesktop or an array of PrivateDesktop objects. You can also use the Uid or an array of Uids instead.

You cannot modify many properties of a private desktop as these contain status information; for example DNSName, RegistrationState, and OSVersion.

Use Add- and Remove- cmdlets to update relationships between private desktops and other objects. For example, you can add a tag to a private desktop with:

```
Add-BrokerTag $tag -Desktop $desktop.Uid
```

Similarly, assign users to private desktops with:

```
Add-BrokerUser $user -PrivateDesktop $desktop
```

Many of the fields that can be set with this cmdlet can also be set with Set-BrokerMachine, such as MaintenanceMode. Using Set-BrokerMachine is preferred in these cases.

For more information about desktops, see about\_Broker\_Desktops; for more information about machines, see about\_Broker\_Machines.

## Related topics

[Get-BrokerMachine](#)

## Parameters

**-InputObject** <PrivateDesktop[]>

Specifies the desktop or array of desktops to modify. You can also use an integer Uid of the desktop instead.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-MachineName**<String>

Specifies the desktop to modify using its machine name (in the form 'domain\machine').

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-AssignedClient Name**<String>

Changes the client name assignment of the desktop. Set this to \$null to remove the assignment. Desktops can be assigned to multiple users, a single IP address, or a single client name, but only to one of these categories at one time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AssignedIPAddress**<String>

Changes the IP address assignment of the desktop. Set this to \$null to remove the assignment. Desktops can be assigned to multiple users, a single IP address, or a single client name, but only to one of these categories at one time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ColorDepth**<ColorDepth>

Changes the color depth connections to this desktop should use.

Valid values are \$null, FourBit, EightBit, SixteenBit, and TwentyFourBit. A value of \$null results in the desktop group value being used instead.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Changes the description of the desktop. This is seen only by Citrix Administrators and is not visible to users.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IconUid**<Int32>

Changes the icon displayed for this desktop. When this setting is \$null, the icon displayed is determined by the desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InMaintenanceMode**<Boolean>

Changes the maintenance mode setting of a desktop. When a desktop is in maintenance mode, it is not included as a candidate when brokering new sessions, and it does not participate in automatic power management (idle pool); however, it still responds to explicit power operations.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PublishedName**<String>

Changes the name displayed to the user for this desktop. When this setting is \$null, the name displayed is determined by the PublishedName of the desktop group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaRequired**<Boolean>

Changes whether or not SecureICA is required for connections to this desktop. When this setting is \$null, the SecureIcaRequired setting from the desktop group is used.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.PrivateDesktop You can pipe PrivateDesktop objects into this cmdlet instead of on the command line with the -InputObject parameter.

**Return Values**

None or Citrix.Broker.Admin.SDK.PrivateDesktop

This cmdlet does not generate any output, unless you use the PasThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.PrivateDesktop object.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Set-BrokerPrivateDesktop DOMAIN\Machine1 -ColorDepth SixteenBit  
Change the color depth of Machine1 to be 16-bit.

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerPrivateDesktop -InMaintenanceMode \$true | Set-BrokerPrivateDesktop -InMaintenanceMode \$false  
Bring all private desktops currently in maintenance mode back into normal service.

# Set-BrokerRebootCycleMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for RebootCycle

## Syntax

```
Set-BrokerRebootCycleMetadata [-RebootCycleId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerRebootCycleMetadata [-RebootCycleId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerRebootCycleMetadata [-RebootCycleId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerRebootCycleMetadata [-InputObject] <RebootCycle[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerRebootCycleMetadata [-InputObject] <RebootCycle[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerRebootCycleMetadata cmdlet creates/updates metadata key-value pairs for RebootCycle. The RebootCycle can be specified by InputObject or piping.

## Related topics

### Parameters

**-RebootCycleId**<Int64>

Specifies the RebootCycle object whose Metadata is to be created/updated by ID.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-InputObject**<RebootCycle[]>

Specifies the RebootCycle objects whose Metadata is to be created/updated.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                |
|------------------------|----------------|
| Accept Pipeline Input? | true (ByValue) |
|------------------------|----------------|

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | False |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.BrokerRebootCycle You can pipe the RebootCycle to hold the new or updated metadata.

**Return Values**

None or Citrix.Broker.Admin.SDK.BrokerRebootCycle

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerRebootCycle object.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerRebootCycleMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the RebootCycle whose instance is pointed by \$obj-Uid

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerRebootCycle | Set-BrokerRebootCycleMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the RebootCycle in the site

# Set-BrokerRebootSchedule

Jan 04, 2017

Updates the values of one or more desktop group reboot schedules.

## Syntax

```
Set-BrokerRebootSchedule [-InputObject] <RebootSchedule[]> [-PassThru] [-Day <RebootScheduleDays>] [-Enabled <Boolean>] [-Frequency <RebootScheduleFrequency>] [-RebootDuration <Int32>] [-StartTime <TimeSpan>] [-WarningDuration <Int32>] [-WarningMessage <String>] [-WarningTitle <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerRebootSchedule [-DesktopGroupName] <String> [-PassThru] [-Day <RebootScheduleDays>] [-Enabled <Boolean>] [-Frequency <RebootScheduleFrequency>] [-RebootDuration <Int32>] [-StartTime <TimeSpan>] [-WarningDuration <Int32>] [-WarningMessage <String>] [-WarningTitle <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerRebootSchedule cmdlet is used to alter the settings of an existing desktop group reboot schedule.

## Related topics

[Get-BrokerRebootSchedule](#)

[New-BrokerRebootSchedule](#)

[Remove-BrokerRebootSchedule](#)

## Parameters

**-Input Object** <RebootSchedule[]>

The reboot schedule to be modified.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-DesktopGroupName** <String>

The name of the desktop group whose reboot schedule is to be modified.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru** <SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-Day** <RebootScheduleDays>

For weekly schedules, the day of the week on which the scheduled reboot-cycle starts (one of Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Boolean that indicates if the reboot schedule is to be enabled or disabled.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Frequency**<RebootScheduleFrequency>

Frequency with which this schedule runs (either Weekly or Daily).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Reboot Duration**<Int32>

Approximate maximum number of minutes over which the scheduled reboot cycle runs.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartTime**<TimeSpan>

Time of day at which the scheduled reboot cycle starts (HH:MM).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WarningDuration**<Int32>

Time prior to the initiation of a machine reboot at which warning message is displayed in all user sessions on that machine. If the warning duration is zero then no message is displayed.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WarningMessage<String>**

Warning message displayed in user sessions on a machine scheduled for reboot. If the message is blank then no message is displayed.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WarningTitle<String>**

The window title used when showing the warning message in user sessions on a machine scheduled for reboot.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId<Guid>**

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.RebootSchedule Reboot schedules may be specified through pipeline input.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Set-BrokerRebootSchedule -DesktopGroupName Accounting -WarningMessage "Save your work" -WarningDuration 10 -WarningTitle "WARNING: Reboot pending"  
Sets the reboot schedule for the desktop group named Accounting to display a message with the title "WARNING: Reboot pending" and body "Save your work" ten minutes prior to rebooting each machine. The message is displayed in every user session on that machine.

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerRebootSchedule -Frequency Weekly | Set-BrokerRebootSchedule -Day Friday  
Sets all weekly reboot schedules to run on Friday.



# Set-BrokerRemotePCAccount

Jan 04, 2017

Modify one or more RemotePCAccounts.

## Syntax

```
Set-BrokerRemotePCAccount [-InputObject] <RemotePCAccount[]> [-PassThru] [-AllowSubfolderMatches <Boolean>] [-MachinesExcluded <String[]>] [-MachinesIncluded <String[]>] [-OU <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Modify one or more RemotePCAccounts.

## Related topics

[Get-BrokerRemotePCAccount](#)

[New-BrokerRemotePCAccount](#)

[Remove-BrokerRemotePCAccount](#)

## Parameters

**-InputObject**<RemotePCAccount[]>

Specifies the RemotePCAccounts to modify.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-AllowSubfolderMatches**<Boolean>

When true a machine matches this RemotePCAccount if the AD computer is in the container specified by the OU property, or within a child container of the OU.

When false the AD computer object only matches if it is directly in the AD container specified by the OU property.

This property is not meaningful when OU has the special value 'any'.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-MachinesExcluded**<String[]>

MachinesExcluded specifies a set of strings that can include asterisk wildcards. If a machine name matches any entries in MachinesExcluded then it cannot match with this RemotePCAccount regardless of whether there is a MachinesIncluded match.

Matches are performed against the domain name joined with the machine name by a backslash (DOMAIN\MACHINE), e.g.:

DOMAIN1\M\*

DOMAIN\*\M\*

\*\M\*

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-MachinesIncluded**<String[]>

MachinesIncluded specifies a set of strings that can include asterisk wildcards. A machine may only match with this RemotePCAccount if it matches a MachinesIncluded entry and does not match any MachinesExcluded entries.

Matches are performed against the domain name joined with the machine name by a backslash (DOMAIN\MACHINE), e.g.:

DOMAIN1\M\*

DOMAIN\*\M\*

\*\M\*

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OU**<String>

Specifies the DN of an AD container, or has the special value 'any'.

When an AD container is specified a machine may only match with the RemotePCAccount when the AD computer object is located relative to the OU.

When 'any' is specified the location of the AD computer object is ignored for purposes of matching this RemotePCAccount. The machine must still meet the MachinesIncluded and MachinesExcluded filters for a match to occur.

In the event that a machine matches with multiple RemotePCAccounts then the RemotePCAccount OU with the longest canonical name takes precedence. The special 'any' OU is treated as lowest priority.

Note that the OU value of every RemotePCAccount must be unique, and this includes only one 'any' entry being permitted.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.RemotePCAccount You can pipe the RemotePCAccounts to be modified into this cmdlet.

### Return Values

None or Citrix.Broker.Admin.SDK.RemotePCAccount

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.RemotePCAccount object.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerRemotePCAccount | Set-BrokerRemotePCAccount -MachinesExcluded @('DOMAIN42\*')  
Make all RemotePCAccounts filter out machines from DOMAIN42.
```

# Set-BrokerSession

Jan 04, 2017

Sets properties of a session.

## Syntax

```
Set-BrokerSession [-InputObject] <Session[]> [-PassThru] [-Hidden <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSession [-SessionKey] <Guid> [-PassThru] [-Hidden <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerSession cmdlet sets properties on a session or set of sessions. You can specify a single session by Uid or SessionKey (Guid) or multiple session instances can be passed to the command by piping or using the -InputObject parameter.

## Related topics

[Get-BrokerSession](#)

## Parameters

**-InputObject**<Session[]>

The session instances whose properties you want to set.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-SessionKey**<Guid>

The session key (Guid) of the session whose properties you want to set.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-Hidden**<Boolean>

Changes whether the session is hidden or not. Hidden sessions are treated as though they do not exist when brokering sessions; a hidden session cannot be reconnected to, but a new session may be launched using the same entitlement.

A session may be hidden automatically by the system when an attempt is made to reconnect to a session that is unreachable due to, for example, the failure of a VM's hypervisor. This allows a new session to be brokered provided that other machines in the same desktop group are still available. If the original session still exists after the hypervisor is restored then it must be unhidden to allow the user to reconnect to it.

Only sessions for shared desktops or applications can be hidden or unhidden. Private desktop or application sessions can never be hidden.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host

name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.Session You can pipe in the sessions whose properties you want to set.

### Return Values

None or Citrix.Broker.Admin.SDK.Session

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.Session object.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> $sessions = Get-BrokerSession -User ACCOUNTS\JohnSmith -Hidden $true  
C:\PS> $sessions | Set-BrokerSession -Hidden $false
```

Finds all sessions in the site for user John Smith that have been hidden, and makes them available for reconnection again.

# Set-BrokerSessionLinger

Jan 04, 2017

Updates the values of one or more desktop group session linger settings.

## Syntax

```
Set-BrokerSessionLinger [-InputObject] <SessionLinger[]> [-PassThru] [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionLinger [-DesktopGroupName] <String> [-PassThru] [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerSessionLinger cmdlet is used to alter the settings of an existing desktop group session linger setting.

## Related topics

[New-BrokerSessionLinger](#)

[Get-BrokerSessionLinger](#)

[Remove-BrokerSessionLinger](#)

## Parameters

**-InputObject** <SessionLinger[]>

The session linger to be modified.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-DesktopGroupName** <String>

The name of the desktop group whose session linger setting is to be modified.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |



**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Boolean that indicates if the session linger setting is to be enabled or disabled.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxAverageLoadThreshold**<Int32>

Specifies the average load threshold across the desktop group. When the threshold hits, lingering sessions across the group be terminated to reduce load. Sessions that have been lingering the longest will be chosen first.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxLoadPerMachineThreshold**<Int32>

Specifies the maximum load threshold per machine in the desktop group. When the threshold hits, lingering sessions on each loaded machine will be terminated to reduce load. Sessions that have been lingering the longest will be chosen first.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-MaxTimeBeforeDisconnect**<TimeSpan>

Specifies the time by which a lingering session will be disconnected. The disconnect time cannot be greater than the terminate timer (if enabled). When the disconnect and terminate times are the same, the terminate time takes precedence.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxTimeBeforeTerminate**<TimeSpan>

Specifies the time by which a lingering session will be terminated. When the disconnect and terminate times are the same, the terminate time takes precedence.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserFilterEnabled**<Boolean>

Specifies whether the session linger's user filter is enabled or disabled. Where the user filter is enabled, lingering is enabled only to users who appear in the filter (either explicitly or by virtue of group membership).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.SessionLinger Session linger settings may be specified through pipeline input.

**Return Values**

None or Citrix.Broker.Admin.SDK.SessionLinger

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.SessionLinger object.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerSessionLinger -DesktopGroupName Accounting -MaxTimeBeforeDisconnect 0:10
Sets the disconnect time for the session linger setting associated with desktop group named Accounting.
```

# Set-BrokerSessionMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for Session

## Syntax

```
Set-BrokerSessionMetadata [-SessionId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-SessionId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-SessionId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-InputObject] <Session[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-InputObject] <Session[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-SessionName] <Guid> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-SessionName] <Guid> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerSessionMetadata cmdlet creates/updates metadata key-value pairs for Session. The Session can be specified by InputObject or piping.

## Related topics

### Parameters

**-SessionId**<Int64>

Specifies the Session object whose Metadata is to be created/updated by ID.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Session[]>

Specifies the Session objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-SessionName**<Guid>

Specifies the Session object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type

Citrix.Broker.Admin.SDK.BrokerSession You can pipe the Session to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerSession

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerSession object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerSessionMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Session whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerSession | Set-BrokerSessionMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Session in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerSessionMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Session in the site whose name is 'objname'

# Set-BrokerSessionPreLaunch

Jan 04, 2017

Updates the values of one or more desktop group session pre-launch settings.

## Syntax

```
Set-BrokerSessionPreLaunch [-InputObject] <SessionPreLaunch[]> [-PassThru] [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionPreLaunch [-DesktopGroupName] <String> [-PassThru] [-Enabled <Boolean>] [-MaxAverageLoadThreshold <Int32>] [-MaxLoadPerMachineThreshold <Int32>] [-MaxTimeBeforeDisconnect <TimeSpan>] [-MaxTimeBeforeTerminate <TimeSpan>] [-UserFilterEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerSessionPreLaunch cmdlet is used to alter the settings of an existing desktop group session pre-launch setting.

## Related topics

[New-BrokerSessionPreLaunch](#)

[Get-BrokerSessionPreLaunch](#)

[Remove-BrokerSessionPreLaunch](#)

## Parameters

**-InputObject**<SessionPreLaunch[]>

The session pre-launch to be modified.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-DesktopGroupName**<String>

The name of the desktop group whose session pre-launch setting is to be modified.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|



|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-Enabled**<Boolean>

Boolean that indicates if the session pre-launch setting is to be enabled or disabled.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxAverageLoadThreshold**<Int32>

Specifies the average load threshold across the desktop group. When the threshold hits, pre-launched sessions across the group be terminated to reduce load. Sessions that have been pre-launched the longest will be chosen first.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxLoadPerMachineThreshold**<Int32>

Specifies the maximum load threshold per machine in the desktop group. When the threshold hits, pre-launched sessions on each loaded machine will be terminated to reduce load. Sessions that have been pre-launched the longest will be chosen first.

|  |  |
|--|--|
|  |  |
|--|--|

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxTimeBeforeDisconnect**<TimeSpan>

Specifies the time by which a pre-launched session will be disconnected. The disconnect time cannot be greater than the terminate timer (if enabled). When the disconnect and terminate times are the same, the terminate time takes precedence.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-MaxTimeBeforeTerminate**<TimeSpan>

Specifies the time by which a pre-launched session will be terminated. When the disconnect and terminate times are the same, the terminate time takes precedence.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-UserFilterEnabled**<Boolean>

Specifies whether the session pre-launch's user filter is enabled or disabled. Where the user filter is enabled, pre-launch is enabled only to users who appear in the filter (either explicitly or by virtue of group membership).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop

Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.SessionPreLaunch Session pre-launch settings may be specified through pipeline input.

**Return Values**

None or Citrix.Broker.Admin.SDK.SessionPreLaunch

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.SessionPreLaunch object.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerSessionPreLaunch -DesktopGroupName Accounting -MaxTimeBeforeDisconnect 0:10
Sets the disconnect time for the session pre-launch setting associated with desktop group named Accounting.
```

# Set-BrokerSharedDesktop

Jan 04, 2017

Change the settings of a shared desktop.

## Syntax

```
Set-BrokerSharedDesktop [-InputObject] <SharedDesktop[]> [-PassThru] [-InMaintenanceMode <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSharedDesktop [-MachineName] <String> [-PassThru] [-InMaintenanceMode <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Shared desktops are automatically created when a machine is added to a desktop group with a DesktopKind of 'Shared', and these inherit default properties. Use Set-BrokerSharedDesktop to change the configuration settings of an existing shared desktop.

To specify shared desktops, you can choose whether to update by machine name or by passing a SharedDesktop or an array of SharedDesktop objects. You can also use the Uid or an array of Uids instead.

Most properties of a shared desktop cannot be modified as these contain status information; for example DNSName, RegistrationState, and OSVersion. You can change only the maintenance mode setting with this cmdlet.

Many of the properties that can be set with Set-BrokerSharedDesktop can be set by using Set-BrokerMachine (e.g. InMaintenanceMode). Using the Set-BrokerMachine cmdlet, where possible, is the preferred behaviour.

See about\_Broker\_Desktops for more information about desktops.

## Related topics

[Get-BrokerSharedDesktop](#)

## Parameters

**-InputObject**<SharedDesktop[]>

Specifies the desktop or array of desktops to modify. You can also use an integer Uid of the desktop instead.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-MachineName**<String>

Specifies the desktop to modify using its machine name (in the form 'domain\machine').

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-InMaintenanceMode**<Boolean>

Changes the maintenance mode setting of a desktop. When a desktop is in maintenance mode, it is not included as a candidate when brokering new sessions, and it does not participate in automatic power management (idle pool); however, it still responds to explicit power operations.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

Citrix.Broker.Admin.SDK.SharedDesktop You can pipe SharedDesktop objects into this cmdlet instead of on the command line with the -InputObject parameter.

## Return Values

None or Citrix.Broker.Admin.SDK.SharedDesktop

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.SharedDesktop object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerSharedDesktop DOMAIN\Machine1 -InMaintenanceMode $true  
Put the Machine1 shared desktop into maintenance mode.
```

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerSharedDesktop -InMaintenanceMode $true | Set-BrokerSharedDesktop -InMaintenanceMode $false  
Bring all shared desktops currently in maintenance mode back into normal service.
```

# Set-BrokerSite

Jan 04, 2017

Changes the overall settings of the current XenDesktop broker site.

## Syntax

```
Set-BrokerSite [-PassThru] [-BaseOU <Guid>] [-ColorDepth <ColorDepth>] [-ConnectionLeasingEnabled <Boolean>] [-DesktopGroupIconUid <Int32>] [-DnsResolutionEnabled <Boolean>] [-SecureIcaRequired <Boolean>] [-TrustRequestsSentToTheXmlServicePort <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerSite cmdlet modifies properties of the current broker site.

The broker site is a top-level, logical representation of the XenDesktop site, from the perspective of the brokering services running within the site. It defines various site-wide default attributes used by the brokering services.

A XenDesktop installation has only a single broker site instance.

## Related topics

[Get-BrokerSite](#)

[New-BrokerIcon](#)

## Parameters

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-BaseOU**<Guid>

Changes the objectGUID property identifying the base OU in Active Directory used for desktop registrations. For sites using only registry-based discovery (the default) this value is \$null.

Any desktop attempting to register through a different OU from the one specified here is rejected. Note that desktops configured for registry-based discovery can register with the site, even if a BaseOU value is specified.

Information held in Active Directory is not modified by changing this value.

Typically, this property is changed only by using the Set-ADControllerDiscovery.ps1 script.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ColorDepth**<ColorDepth>

Changes the default color depth for new desktop groups, if no color depth is specified explicitly when a group is created. Changing this default has no impact on the color depths used already by existing groups.

Valid values are FourBit, EightBit, SixteenBit, and TwentyFourBit.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ConnectionLeasingEnabled**<Boolean>

Enabled or disable connection leasing on the site.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DesktopGroupIconUid**<Int32>

Changes the default desktop icon used for new desktop groups if no icon is specified explicitly when a group is created. Changing this default has no impact on the icons used already by existing groups.

The specified icon must already have been added to the site using New-BrokerIcon.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-DnsResolutionEnabled**<Boolean>

Changes whether ICA files returned by a broker service to a user device contain the numeric IP address or the DNS name of the desktop machine to which a session should be established.

With the default value (\$false), ICA files will always contain a numeric IP address. To have DNS names appear in the ICA files, set the value to \$true.

Even when DNS resolution is enabled (\$true), IP addresses may still appear in ICA files. The reasons for this include, for example, that the broker service is unable to obtain a DNS name for the target machine, or that Web Interface is configured to always use numeric IP addresses in this context.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SecureIcaRequired**<Boolean>

Changes the default SecureICA usage requirements for new desktop groups if no SecureICA setting is specified explicitly when a group is created. Changing this default has no impact on the SecureICA usage requirements of existing groups.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TrustRequestsSentToTheXmlServicePort**<Boolean>

Changes whether the XML Service (as used by Web Interface) implicitly trusts the originator of requests it receives, or whether it fully authenticates them.

With the default value (\$false), full authentication checks are performed. However, you must enable this setting (\$true) to allow support for "Pass-through" authentication, and/or connections routed through Access Gateway.

If this setting is enabled, you must ensure that controllers running the brokering services are securely firewalled.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

None You cannot pipe input into this cmdlet.

### Return Values

None or Citrix.Broker.Admin.SDK.Site

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.Site object.

### Examples

#### ----- **EXAMPLE 1** -----

```
C:\PS> Set-BrokerSite -ColorDepth SixteenBit
```

Specifies that any new desktop groups created, where a color depth value is not specified, default to using 16-bit color depth for user sessions to desktops or applications within that group.

# Set-BrokerSiteMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for Site

## Syntax

```
Set-BrokerSiteMetadata -Name <String> -Value <String> [[-InputObject] <Site[]>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSiteMetadata -Map <PSObject> [[-InputObject] <Site[]>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSiteMetadata -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSiteMetadata -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSiteMetadata -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerSiteMetadata cmdlet creates/updates metadata key-value pairs for Site. The Site can be specified by InputObject or piping.

## Related topics

### Parameters

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-InputObject**<Site[]>

Specifies the Site objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.BrokerSite You can pipe the Site to hold the new or updated metadata.

**Return Values**

None or Citrix.Broker.Admin.SDK.BrokerSite

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerSite object.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Set-BrokerSiteMetadata -Name "MyMetadataName" -Value "1234"  
This command creates/updates the Metadata "MyMetadataName" key-value pair for the Site

----- **EXAMPLE 2** -----

C:\PS> Get-BrokerSite | Set-BrokerSiteMetadata -Name "MyMetadataName" -Value "1234"  
This command creates/updates metadata key "MyMetadataName" with the value "1234"

# Set-BrokerTagMetadata

Jan 04, 2017

Creates/Updates metadata key-value pairs for Tag

## Syntax

```
Set-BrokerTagMetadata [-TagId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerTagMetadata [-TagId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerTagMetadata [-TagId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerTagMetadata [-InputObject] <Tag[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerTagMetadata [-InputObject] <Tag[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerTagMetadata [-TagName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerTagMetadata [-TagName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-BrokerTagMetadata cmdlet creates/updates metadata key-value pairs for Tag. The Tag can be specified by InputObject or piping.

## Related topics

### Parameters

**-TagId**<Int32>

Specifies the Tag object whose Metadata is to be created/updated by ID.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-InputObject**<Tag[]>

Specifies the Tag objects whose Metadata is to be created/updated.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-TagName**<String>

Specifies the Tag object whose Metadata is to be created/updated by name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Name**<String>

Specifies the name of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Value**<String>

Specifies the value of the Metadata member to be created/updated

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Map**<PSObject>

Specifies a hashtable containing name/value pairs to be used to create or update Metadata members

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-PassThru**<SwitchParameter>

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it returns the affected record.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

Input Type



Citrix.Broker.Admin.SDK.BrokerTag You can pipe the Tag to hold the new or updated metadata.

## Return Values

None or Citrix.Broker.Admin.SDK.BrokerTag

This cmdlet does not generate any output, unless you use the PassThru parameter, in which case it generates a Citrix.Broker.Admin.SDK.BrokerTag object.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Set-BrokerTagMetadata -InputObject $obj-Uid -Name "MyMetadataName" -Value "1234"
```

This command creates/updates the Metadata "MyMetadataName" key-value pair for the Tag whose instance is pointed by \$obj-Uid

### ----- EXAMPLE 2 -----

```
C:\PS> Get-BrokerTag | Set-BrokerTagMetadata -Name "MyMetadataName" -Value "1234"
```

This command creates/updates metadata key "MyMetadataName" with the value "1234" for all the Tag in the site

### ----- EXAMPLE 3 -----

```
C:\PS> @{ 'name1' = 'value1'; 'name2' = 'value2' } | Set-BrokerTagMetadata 'objname'
```

This command creates/updates two metadata keys "name1" and "name2" with the values "value1" and "value2" respectively for the Tag in the site whose name is 'objname'

# Start-BrokerCatalogPvdImagePrepare

Jan 04, 2017

Start the PVD Image prepare process in the Broker for the machines in the specified catalog(s).

## Syntax

```
Start-BrokerCatalogPvdImagePrepare [-InputObject] <Catalog[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

The Start-BrokerCatalogPvdImagePrepare cmdlet instructs the Broker to request the Personal VDisk (PVD) preparation process for all of the machines in the specified catalog(s). The process begins when each machine is subsequently powered off, at which point brokering of user desktop sessions is suspended as well as regular machine power operations until the process completes. Only catalogs with a PersistUserChanges value of OnPvd are supported by this cmdlet.

## Related topics

[Start-BrokerMachinePvdImagePrepare](#)

## Parameters

**-InputObject**<Catalog[]>

The catalog(s) holding the machines on which to start the PVD Image Preparation process.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.Catalog You can pipe in the catalogs on which to start the PVD image preparation process.

### Return Values

None

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerCatalog 'MyCatalog' | Start-BrokerCatalogPvdImagePrepare
```

Instruct the Broker that the Personal VDisk preparation process will start the next time the machines in the specified catalog are powered off.

# Start-BrokerMachinePvdImagePrepare

Jan 04, 2017

Start the PVD Image prepare process in the Broker for the specified machine(s).

## Syntax

```
Start-BrokerMachinePvdImagePrepare [-InputObject] <Machine[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

The Start-BrokerMachinePvdImagePrepare cmdlet instructs the Broker to request the Personal VDisk (PVD) preparation process for the specified machine(s). The process begins when each machine is subsequently powered off, at which point brokering of user desktop sessions is suspended as well as regular machine power operations until the process completes. Only machines in catalogs with a PersistUserChanges value of OnPvd are supported by this cmdlet.

## Related topics

[Start-BrokerCatalogPvdImagePrepare](#)

## Parameters

**-InputObject**<Machine[]>

The machine(s) to start the PVD Image Preparation process on.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.Machine You can pipe in the machines to start the PVD image preparation process on.

### Return Values

None

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerMachine 'MyMachine' | Start-BrokerMachinePvdImagePrepare
```

Instruct the Broker that the Personal VDisk preparation process will start the next time the specified machine(s) are powered off.

# Start-BrokerNaturalRebootCycle

Jan 04, 2017

Reboots all machines from the specified catalog when they are not in use.

## Syntax

```
Start-BrokerNaturalRebootCycle [-InputObject] <Catalog[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Creating a natural reboot cycle for catalog ensures that all machines in the catalog are running the most recent image for the catalog and that all PvD image updates have been performed.

The machines are rebooted in a non disruptive manner, allowing machines that are in use to continue working and be restarted only after they become idle.

## Related topics

None

## Parameters

### **-InputObject**<Catalog[]>

Reboots all machines from this input catalog. The catalogs can be specified using UID values, name values (including wildcards) or catalog objects.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

### **-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.Catalog Catalogs may be specified through pipeline input. The catalogs can be specified using UID values, name values (including wildcards) or catalog objects

### Return Values

None

### Notes

Natural reboot cycles do not apply to non-power managed and multi session catalogs

### Examples

----- **EXAMPLE 1** -----

```
$c = Get-BrokerCatalog -Uid 1
```

```
Start-BrokerNaturalRebootCycle -InputObject $c
```

The above code applies a natural reboot cycle on catalog with Id 1

# Start-BrokerRebootCycle

Jan 04, 2017

Creates and starts a reboot cycle for each desktop group that contains machines from the specified catalog.

## Syntax

```
Start-BrokerRebootCycle [-InputObject] <Catalog[]> -RebootDuration <Int32> [-WarningDuration <Int32>] [-WarningTitle <String>] [-WarningMessage <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Start-BrokerRebootCycle cmdlet is used to create and start a reboot cycle for each desktop group that contains machines from the specified catalog. For a given desktop group, only the machines from the target catalog are rebooted and any machines from other catalogs are not rebooted.

Creating a reboot cycle for a catalog ensures that all machines in the catalog are running the most recent image for the catalog and that all PVD image updates have been performed.

## Related topics

[Stop-BrokerRebootCycle](#)

[Get-BrokerRebootCycle](#)

## Parameters

**-InputObject**<Catalog[]>

Creates a reboot cycle for each desktop group that contains machines from this input catalog SDK object. The catalogs can be specified using UID values, name values (including wildcards) or catalog objects.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-RebootDuration**<Int32>

Approximate maximum duration in minutes over which the reboot cycle runs.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WarningDuration**<Int32>

Time in minutes prior to a machine reboot at which a warning message is displayed in all user sessions on that machine. If the warning duration value is zero then no message is displayed.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |



**-WarningTitle**<String>

The window title used when showing the warning message in user sessions on a machine scheduled for reboot.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-WarningMessage**<String>

Warning message displayed in user sessions on a machine scheduled for reboot. If the message is blank then no message is displayed.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Direct or typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Citrix.Broker.Admin.SDK.Catalog Catalogs may be specified through pipeline input. The catalogs can be specified using UID values, name values (including wildcards) or catalog objects

**Return Values**

none

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerCatalog -Name "SampleCatalog" | Start-BrokerRebootCycle -RebootDuration 240 -WarningMessage "Save your work" -WarningDuration 15
```

Starts a new reboot cycle for each desktop group containing machines from the catalog "SampleCatalog". Each reboot cycle will have a duration of six hours. Fifteen minutes prior to rebooting a machine, the message "Save your work" will be displayed in each active user session.

# Stop-BrokerRebootCycle

Jan 04, 2017

Cancels the specified reboot cycle.

## Syntax

```
Stop-BrokerRebootCycle [-InputObject] <RebootCycle[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

The Stop-BrokerRebootCycle cmdlet is used to cancel the specified reboot cycle.

## Related topics

[Get-BrokerRebootCycle](#)

[Start-BrokerRebootCycle](#)

## Parameters

**-InputObject**<RebootCycle[]>

Cancels this reboot cycle.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host

name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

### Input Type

Citrix.Broker.Admin.SDK.RebootCycle Reboot cycles may be specified through pipeline input.

### Return Values

none

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerRebootCycle -CatalogUid 7 | Stop-BrokerRebootCycle  
Cancels every reboot cycle for the catalog that has the Uid of 7.
```

# Stop-BrokerSession

Jan 04, 2017

Stop or log off a session.

## Syntax

```
Stop-BrokerSession [-InputObject] <Session[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Stops or logs off sessions.

## Related topics

[Disconnect-BrokerSession](#)

[Get-BrokerSession](#)

## Parameters

**-InputObject**<Session[]>

Identifies the session(s) to terminate. This can be expressed as either a session Uid or a session object.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

Citrix.Broker.Admin.SDK.Session The sessions to stop can be piped into this cmdlet.

## Return Values

None

## Notes

This operation is non-blocking and returns before it completes. The operation, however, is unlikely to fail unless there are communication problems between the controller and the machine, if bad arguments are passed to the cmdlet itself or if the machine cannot successfully execute the operation.

The transient nature of sessions means that the list of session objects or UIDs supplied to Stop-BrokerSession could consist of valid and invalid sessions. Invalid sessions are detected and disregarded and the stop session operation is invoked on the valid sessions.

The system can fail to invoke the operation if the machine is not in an appropriate state or if there are problems in communicating with the machine. When an operation is invoked the system detects if the operation was initiated successfully or not by the machine. As this operation is non-blocking the system doesn't detect or report whether the operation ultimately succeeded or failed after its successful initialization on the machine.

Operation failures are reported through the broker SDK error handling mechanism (see about\_Broker\_ErrorHandling). In the event of errors the SdkErrorRecord error status is set to SessionOperationFailed and its error data dictionary is populated with the following entries:

- o OperationsAttemptedCount - The number of operations attempted.
- o OperationsFailedCount - The number of failed operations.
- o OperationsSucceededCount - The number of successfully executed operations.
- o UnresolvedSessionFailuresCount - The number of operations that failed due to invalid sessions being supplied.
- o OperationInvocationFailuresCount - The number of operations that failed because they could not be invoked on the desktop.
- o DesktopExecutionFailuresCount - The number of operations that failed because they could not be successfully executed by the desktop.

The SdkErrorRecord message will also display the number of attempted, failed and successful operations in the following format:

```
"Session operation error - attempted:<OperationsAttemptedCount>, failed:<OperationsFailedCount>, succeeded:
<OperationsSucceededCount>"
```

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerSession -UserName MyDomain\MyAccount | Stop-BrokerSession
Stops all sessions for the user MyDomain\MyAccount.
```

### ----- EXAMPLE 2 -----

```
C:\PS> $desktop = Get-BrokerDesktop -DNSName MyMachine.MyDomain.com
C:\PS> Stop-BrokerSession $desktop.SessionUid
Stops the session on MyMachine.
```

### ----- EXAMPLE 3 -----

```
C:\PS> Get-BrokerSession -Filter { SessionState -eq 'Disconnected' -and SessionStateChangeTime -lt '-1' } | Stop-BrokerSession
Stop sessions that have been disconnected for more than one day.
```

### ----- EXAMPLE 4 -----

```
C:\PS> trap [Citrix.Broker.Admin.SDK.SdkOperationException]
C:\PS> {
C:\PS> write $("Exception name = " + $_.Exception.GetType().FullName)
C:\PS> write $("SdkOperationException.Status = " + $_.Exception.Status)
C:\PS> write $("SdkOperationException.ErrorData=")
C:\PS> $_.Exception.ErrorData
C:\PS>
C:\PS> write $("SdkOperationException.InnerException = " + $_.Exception.InnerException)
C:\PS> $_.Exception.InnerException
C:\PS> continue
C:\PS> }
C:\PS>
C:\PS> Stop-BrokerSession -InputObject 10,11,12
Trap and display error information.
```

# Test-BrokerAccessPolicyRuleNameAvailable

Jan 04, 2017

Determine whether the proposed AccessPolicyRule Name is available for use.

## Syntax

```
Test-BrokerAccessPolicyRuleNameAvailable [-Name] <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet checks whether proposed AccessPolicyRule Name is available for use. It returns a record for each Name indicating the availability of that Name, with \$true indicating that the Name is unused and available for use, or \$false if it is not available.

## Related topics

[Get-BrokerAccessPolicyRule](#)

[New-BrokerAccessPolicyRule](#)

[Rename-BrokerAccessPolicyRule](#)

## Parameters

**-Name**<String[]>

The AccessPolicyRule Name to be tested.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type



System.String You can pipe a string that contains the Name to test.

## Return Values

Citrix.Broker.Admin.SDK.NameAvailability

The cmdlet returns a result for each Name specified. An availability of "True" indicates the Name is available for use, and "False" if it is not available.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Test-BrokerAccessPolicyRuleNameAvailable -Name Test1
```

Checks whether the Name "Test1" is available.

### ----- EXAMPLE 2 -----

```
C:\PS> Test-BrokerAccessPolicyRuleNameAvailable @"Test1","Test2","Test3"
```

Checks whether each of the specified names is available.

# Test-BrokerAppAssignmentPolicyRuleNameAvailable

Jan 04, 2017

Determine whether the proposed AppAssignmentPolicyRule Name is available for use.

## Syntax

```
Test-BrokerAppAssignmentPolicyRuleNameAvailable [-Name] <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet checks whether proposed AppAssignmentPolicyRule Name is available for use. It returns a record for each Name indicating the availability of that Name, with \$true indicating that the Name is unused and available for use, or \$false if it is not available.

## Related topics

[Get-BrokerAppAssignmentPolicyRule](#)

[New-BrokerAppAssignmentPolicyRule](#)

[Rename-BrokerAppAssignmentPolicyRule](#)

## Parameters

**-Name**<String[]>

The AppAssignmentPolicyRule Name to be tested.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

System.String You can pipe a string that contains the Name to test.

## Return Values

Citrix.Broker.Admin.SDK.NameAvailability

The cmdlet returns a result for each Name specified. An availability of "True" indicates the Name is available for use, and "False" if it is not available.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Test-BrokerAppAssignmentPolicyRuleNameAvailable -Name Test1
```

Checks whether the Name "Test1" is available.

### ----- EXAMPLE 2 -----

```
C:\PS> Test-BrokerAppAssignmentPolicyRuleNameAvailable @"Test1","Test2","Test3"
```

Checks whether each of the specified names is available.

# Test-BrokerAppEntitlementPolicyRuleNameAvailable

Jan 04, 2017

Determine whether the proposed AppEntitlementPolicyRule Name is available for use.

## Syntax

```
Test-BrokerAppEntitlementPolicyRuleNameAvailable [-Name] <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet checks whether proposed AppEntitlementPolicyRule Name is available for use. It returns a record for each Name indicating the availability of that Name, with \$true indicating that the Name is unused and available for use, or \$false if it is not available.

## Related topics

[Get-BrokerAppEntitlementPolicyRule](#)

[New-BrokerAppEntitlementPolicyRule](#)

[Rename-BrokerAppEntitlementPolicyRule](#)

## Parameters

**-Name**<String[]>

The AppEntitlementPolicyRule Name to be tested.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

System.String You can pipe a string that contains the Name to test.

## Return Values

Citrix.Broker.Admin.SDK.NameAvailability

The cmdlet returns a result for each Name specified. An availability of "True" indicates the Name is available for use, and "False" if it is not available.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Test-BrokerAppEntitlementPolicyRuleNameAvailable -Name Test1
```

Checks whether the Name "Test1" is available.

### ----- EXAMPLE 2 -----

```
C:\PS> Test-BrokerAppEntitlementPolicyRuleNameAvailable @("Test1","Test2","Test3")
```

Checks whether each of the specified names is available.

# Test-BrokerApplicationNameAvailable

Jan 04, 2017

Determine whether the proposed Application Name is available for use.

## Syntax

```
Test-BrokerApplicationNameAvailable [-Name] <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet checks whether proposed Application Name is available for use. It returns a record for each Name indicating the availability of that Name, with \$true indicating that the Name is unused and available for use, or \$false if it is not available.

## Related topics

[Get-BrokerApplication](#)

[New-BrokerApplication](#)

[Rename-BrokerApplication](#)

## Parameters

**-Name**<String[]>

The Application Name to be tested.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

System.String You can pipe a string that contains the Name to test.

## Return Values

Citrix.Broker.Admin.SDK.NameAvailability

The cmdlet returns a result for each Name specified. An availability of "True" indicates the Name is available for use, and "False" if it is not available.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Test-BrokerApplicationNameAvailable -Name Test1
```

Checks whether the Name "Test1" is available.

### ----- EXAMPLE 2 -----

```
C:\PS> Test-BrokerApplicationNameAvailable @"Test1","Test2","Test3"
```

Checks whether each of the specified names is available.

# Test-BrokerAssignmentPolicyRuleNameAvailable

Jan 04, 2017

Determine whether the proposed AssignmentPolicyRule Name is available for use.

## Syntax

```
Test-BrokerAssignmentPolicyRuleNameAvailable [-Name] <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet checks whether proposed AssignmentPolicyRule Name is available for use. It returns a record for each Name indicating the availability of that Name, with \$true indicating that the Name is unused and available for use, or \$false if it is not available.

## Related topics

[Get-BrokerAssignmentPolicyRule](#)

[New-BrokerAssignmentPolicyRule](#)

[Rename-BrokerAssignmentPolicyRule](#)

## Parameters

**-Name**<String[]>

The AssignmentPolicyRule Name to be tested.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type



System.String You can pipe a string that contains the Name to test.

## Return Values

Citrix.Broker.Admin.SDK.NameAvailability

The cmdlet returns a result for each Name specified. An availability of "True" indicates the Name is available for use, and "False" if it is not available.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Test-BrokerAssignmentPolicyRuleNameAvailable -Name Test1
```

Checks whether the Name "Test1" is available.

### ----- EXAMPLE 2 -----

```
C:\PS> Test-BrokerAssignmentPolicyRuleNameAvailable @"Test1","Test2","Test3"
```

Checks whether each of the specified names is available.

# Test-BrokerCatalogNameAvailable

Jan 04, 2017

Determine whether the proposed Catalog Name is available for use.

## Syntax

```
Test-BrokerCatalogNameAvailable [-Name] <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet checks whether proposed Catalog Name is available for use. It returns a record for each Name indicating the availability of that Name, with \$true indicating that the Name is unused and available for use, or \$false if it is not available.

## Related topics

[Get-BrokerCatalog](#)

[New-BrokerCatalog](#)

[Rename-BrokerCatalog](#)

## Parameters

**-Name**<String[]>

The Catalog Name to be tested.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

System.String You can pipe a string that contains the Name to test.

## Return Values

Citrix.Broker.Admin.SDK.NameAvailability

The cmdlet returns a result for each Name specified. An availability of "True" indicates the Name is available for use, and "False" if it is not available.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Test-BrokerCatalogNameAvailable -Name Test1
```

Checks whether the Name "Test1" is available.

----- **EXAMPLE 2** -----

```
C:\PS> Test-BrokerCatalogNameAvailable @"Test1","Test2","Test3"
```

Checks whether each of the specified names is available.

# Test-BrokerDBConnection

Jan 04, 2017

Tests whether a database is suitable for use by the Citrix Broker Service.

## Syntax

```
Test-BrokerDBConnection [-DBConnection] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Tests whether the database specified in the given connection string is suitable for use by the currently selected Citrix Broker Service instance.

The service attempts to contact the specified database and returns a status indicating whether the database is both contactable and usable. The test does not impact any currently established connection from the service instance to another database in any way. The tested connection string is not recorded.

Only use of Windows authentication within the connection string is supported; a connection string containing SQL authentication credentials is always rejected as invalid.

The current service instance is the one on the local machine, or the one most recently specified using the `-AdminAddress` parameter of a Broker SDK cmdlet.

## Related topics

[Get-BrokerServiceStatus](#)

[Get-BrokerDBConnection](#)

[Set-BrokerDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be tested by the currently selected Citrix Broker Service instance.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Test-BrokerDBConnection cmdlet returns an object describing the status of the selected Broker Service instance that would result if the connection string were used with the Set-BrokerDBConnection cmdlet together with extra diagnostics information for the specified connection string. The actual current status of the service is not changed. Possible diagnostic values are:

-- OK:

The Set-BrokerDBConnection command would succeed if it were executed with the supplied connection string.

-- DBUnconfigured:

No database connection string is set for the service instance.

-- DBRejectedConnection:

The database rejected the logon attempt from the Broker Service instance. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

-- InvalidDBConfigured:

The specified database does not exist, is not visible to the Broker Service instance, or the service's schema within the database is invalid.

-- DBNotFound:

The specified database could not be located with the given connection string.

-- DBNewerVersionThanService:

The Broker Service instance is older than, and incompatible with, the service's schema in the database. The service instance needs upgrading.

-- DBOlderVersionThanService:

The Broker Service instance is newer than, and incompatible with, the service's schema in the database. The database schema needs upgrading.

-- DBVersionChangeInProgress:

A database schema upgrade is currently in progress.

-- PendingFailure:

Connectivity between the Broker Service instance and the database has been lost. This may be a transitory network error, but may indicate a loss of connectivity that requires administrator intervention.

-- Failed:

Connectivity between the Broker Service instance and the database has been lost for an extended period of time, or has failed due to a configuration problem. The service instance cannot operate while its connection to the database is unavailable.

-- Unknown:

Service status cannot be determined.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidDBConnectionString

The database connection string has an invalid format.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS>Test-BrokerDBConnection "Server=dbserver\SQLEXPRESS;Database=XDDB;Trusted_Connection=True"
```

Tests whether the service instance could use a database called XDDB on an SQL Server Express database running on the machine called dbserver. Integrated Windows authentication is required.

# Test-BrokerDesktopGroupNameAvailable

Jan 04, 2017

Determine whether the proposed DesktopGroup Name is available for use.

## Syntax

```
Test-BrokerDesktopGroupNameAvailable [-Name] <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet checks whether proposed DesktopGroup Name is available for use. It returns a record for each Name indicating the availability of that Name, with \$true indicating that the Name is unused and available for use, or \$false if it is not available.

## Related topics

[Get-BrokerDesktopGroup](#)

[New-BrokerDesktopGroup](#)

[Rename-BrokerDesktopGroup](#)

## Parameters

**-Name**<String[]>

The DesktopGroup Name to be tested.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

System.String You can pipe a string that contains the Name to test.

## Return Values

Citrix.Broker.Admin.SDK.NameAvailability

The cmdlet returns a result for each Name specified. An availability of "True" indicates the Name is available for use, and "False" if it is not available.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Test-BrokerDesktopGroupNameAvailable -Name Test1
```

Checks whether the Name "Test1" is available.

### ----- EXAMPLE 2 -----

```
C:\PS> Test-BrokerDesktopGroupNameAvailable @("Test1","Test2","Test3")
```

Checks whether each of the specified names is available.



# Test-BrokerEntitlementPolicyRuleNameAvailable

Jan 04, 2017

Determine whether the proposed EntitlementPolicyRule Name is available for use.

## Syntax

```
Test-BrokerEntitlementPolicyRuleNameAvailable [-Name] <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet checks whether proposed EntitlementPolicyRule Name is available for use. It returns a record for each Name indicating the availability of that Name, with \$true indicating that the Name is unused and available for use, or \$false if it is not available.

## Related topics

[Get-BrokerEntitlementPolicyRule](#)

[New-BrokerEntitlementPolicyRule](#)

[Rename-BrokerEntitlementPolicyRule](#)

## Parameters

**-Name**<String[]>

The EntitlementPolicyRule Name to be tested.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

System.String You can pipe a string that contains the Name to test.

## Return Values

Citrix.Broker.Admin.SDK.NameAvailability

The cmdlet returns a result for each Name specified. An availability of "True" indicates the Name is available for use, and "False" if it is not available.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Test-BrokerEntitlementPolicyRuleNameAvailable -Name Test1
```

Checks whether the Name "Test1" is available.

### ----- EXAMPLE 2 -----

```
C:\PS> Test-BrokerEntitlementPolicyRuleNameAvailable @"Test1","Test2","Test3"
```

Checks whether each of the specified names is available.

# Test-BrokerLicenseServer

Jan 04, 2017

Tests whether or not a license server can be used by the broker.

## Syntax

```
Test-BrokerLicenseServer [-ComputerName] <String> [-AdminAddress <String>] [[-Port] <Int32>] [<CommonParameters>]
```

## Detailed Description

Tests whether or not a given license server can be used by the broker.

## Related topics

[Get-BrokerSite](#)

[Set-BrokerSite](#)

## Parameters

**-ComputerName**<String>

The name of the license server to test (machine.domain).

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          | None  |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**-Port**<Int32>

The port number to use on the server.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          | 27000 |
| Accept Pipeline Input? | false |

### Input Type

None You cannot pipe input into this cmdlet.

### Return Values

System.string

Test-BrokerLicenseServer returns:

- o 'Compatible' - the server is a compatible license server that can be used.
- o 'Incompatible' - the server is an incompatible license server that can't be used.
- o 'Inaccessible' - the server cannot be accessed. The server may be down, unreachable, or non-existent.
- o 'InternalError' - the server can't be used due to an internal error. A required licensing component on the server may not be installed, configured, or working correctly.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Test-BrokerLicenseServer -LicenseServerAddress "machine.domain" 1234
Tests whether or not the license server "machine.domain" with port number 1234 can be used.
```

----- **EXAMPLE 2** -----

```
C:\PS> Test-BrokerLicenseServer -LicenseServerAddress "machine.domain"
Tests whether or not the license server "machine.domain" with port number 2700 can be used.
```

# Test-BrokerMachineNameAvailable

Jan 04, 2017

Determine whether the proposed Machine MachineName is available for use.

## Syntax

```
Test-BrokerMachineNameAvailable [-MachineName] <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet checks whether proposed Machine MachineName is available for use. It returns a record for each MachineName indicating the availability of that MachineName, with \$true indicating that the MachineName is unused and available for use, or \$false if it is not available.

## Related topics

[Get-BrokerMachine](#)

[New-BrokerMachine](#)

## Parameters

**-MachineName**<String[]>

The Machine MachineName to be tested.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

System.String You can pipe a string that contains the MachineName to test.

## Return Values

Citrix.Broker.Admin.SDK.NameAvailability

The cmdlet returns a result for each MachineName specified. An availability of "True" indicates the MachineName is available for use, and "False" if it is not available.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Test-BrokerMachineNameAvailable -MachineName Test1
```

Checks whether the MachineName "Test1" is available.

----- **EXAMPLE 2** -----

```
C:\PS> Test-BrokerMachineNameAvailable @"Test1","Test2","Test3"
```

Checks whether each of the specified names is available.

# Test-BrokerPowerTimeSchemeNameAvailable

Jan 04, 2017

Determine whether the proposed PowerTimeScheme Name is available for use.

## Syntax

```
Test-BrokerPowerTimeSchemeNameAvailable [-Name] <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet checks whether proposed PowerTimeScheme Name is available for use. It returns a record for each Name indicating the availability of that Name, with \$true indicating that the Name is unused and available for use, or \$false if it is not available.

## Related topics

[Get-BrokerPowerTimeScheme](#)

[New-BrokerPowerTimeScheme](#)

[Rename-BrokerPowerTimeScheme](#)

## Parameters

**-Name**<String[]>

The PowerTimeScheme Name to be tested.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

System.String You can pipe a string that contains the Name to test.

## Return Values

Citrix.Broker.Admin.SDK.NameAvailability

The cmdlet returns a result for each Name specified. An availability of "True" indicates the Name is available for use, and "False" if it is not available.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Test-BrokerPowerTimeSchemeNameAvailable -Name Test1
```

Checks whether the Name "Test1" is available.

### ----- EXAMPLE 2 -----

```
C:\PS> Test-BrokerPowerTimeSchemeNameAvailable @"Test1","Test2","Test3"
```

Checks whether each of the specified names is available.



# Test-BrokerRemotePCAccountNameAvailable

Jan 04, 2017

Determine whether the proposed RemotePCAccount OU is available for use.

## Syntax

```
Test-BrokerRemotePCAccountNameAvailable [-OU] <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet checks whether proposed RemotePCAccount OU is available for use. It returns a record for each OU indicating the availability of that OU, with \$true indicating that the OU is unused and available for use, or \$false if it is not available.

## Related topics

[Get-BrokerRemotePCAccount](#)

[New-BrokerRemotePCAccount](#)

## Parameters

**-OU**<String[]>

The RemotePCAccount OU to be tested.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

## Input Type

System.String You can pipe a string that contains the OU to test.

## Return Values

Citrix.Broker.Admin.SDK.NameAvailability

The cmdlet returns a result for each OU specified. An availability of "True" indicates the OU is available for use, and "False" if it is not available.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Test-BrokerRemotePCAccountNameAvailable -OU Test1
```

Checks whether the OU "Test1" is available.

### ----- EXAMPLE 2 -----

```
C:\PS> Test-BrokerRemotePCAccountNameAvailable @("Test1","Test2","Test3")
```

Checks whether each of the specified names is available.

# Update-BrokerImportedFTA

Jan 04, 2017

Imports or updates all of the file type associations for the specified worker.

## Syntax

```
Update-BrokerImportedFTA -DesktopUids <Int32[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Imports or updates the file type associations from a specified worker machine.

File type association associates a file extension (such as ".txt") with an application (such as Notepad). In a Citrix environment file type associations on a user device can be configured so that when an user clicks on a document it launches the appropriate published application. This is known as "content redirection".

Imported file type associations are different from configured file type associations. Imported file type associations are lists of known file type associations for a given desktop group. Configured file type associations are those that are actually associated with published applications for the purposes of content redirection.

Initially the system is not aware of any extensions, and they must be imported by the Citrix administrator. To import or update file type associations from a worker machine, two criteria must be met:

- o The worker machine must not be in use
- o The worker machine must be in maintenance mode

For more information about putting a worker machine in maintenance mode, see the Set-BrokerPrivateDesktop and Set-BrokerSharedDesktop cmdlets.

Imported file type associations are grouped together based on the desktop group of the machine from which they were imported. All file types for a desktop group are deleted. There is no mechanism for deleting a subset imported file type associations for a specific desktop group.

If file type associations are imported more than once for a desktop group, for example, if this cmdlet is run twice for two workers belonging to the same desktop group, all existing imported file type associations for that desktop group are deleted and imported again.

## Related topics

[Get-BrokerImportedFTA](#)

[Remove-BrokerImportedFTA](#)

## Parameters

**-DesktopUids**<Int32[]>

Imports or updates the file type associations from the specified desktop. The desktop must belong to a desktop group of the Private or Shared desktop kind.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

Int32[] An array of Uids for desktops can be supplied as input.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> $desktop = Get-BrokerSharedDesktop -MachineName "ACME\Worker1"
C:\PS> Set-BrokerSharedDesktop $desktop -InMaintenanceMode $true
C:\PS> Update-BrokerImportedFTA -DesktopUids $desktop.Uid
C:\PS> Set-BrokerSharedDesktop $desktop -InMaintenanceMode $false
```

Gets an object for the worker machine named "Worker1" in the "ACME" domain, and ensures no users can connect to it, before importing the file type associations from that desktop and re-enabling it.

# Update-BrokerLocalLeaseCache

Jan 04, 2017

Flushes the local lease cache.

## Syntax

```
Update-BrokerLocalLeaseCache [-Workers] [-Applications] [-Icons] [-Desktops] [-Leases] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Removes all local cached lease data and any state information stored in the registry.

## Related topics

[Remove-BrokerLease](#)

## Parameters

**-Workers**<SwitchParameter>

Removes all locally cached workers on the controller. The worker cache will be repopulated from the current site database contents after a short delay.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-Applications**<SwitchParameter>

Removes all locally cached applications on the controller. The application cache will be repopulated from the current site database contents after a short delay.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-Icons**<SwitchParameter>

Removes all locally cached icons on the controller. The icon cache will be repopulated from the current site database contents after a short delay.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-Desktops**<SwitchParameter>

Removes all locally cached desktops on the controller. The desktop cache will be repopulated from the current site database contents after a short delay.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-Leases**<SwitchParameter>

Removes all locally cached leases on the controller. The lease cache will be repopulated from the current site database contents after a short delay.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high level operation that this cmdlet call forms a part of. Desktop Studio and Desktop Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

#### Input Type

None

#### Return Values

None

#### Notes

The local cache for lease and other data like worker, desktop, application and icon information is removed and will be repopulated from the current site database contents after a short delay.

#### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Update-BrokerLocalLeaseCache
```

Flushes the local lease cache for all objects and deletes any state information stored in the registry.

----- **EXAMPLE 2** -----

```
C:\PS> Update-BrokerLocalLeaseCache -Workers
```

Flushes the local lease cache for all workers and deletes any state information stored in the registry.



# Update-BrokerNameCache

Jan 04, 2017

Performs administrative operations on the user and machine name cache.

## Syntax

```
Update-BrokerNameCache [-Machines] [-Users] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Triggers an immediate asynchronous refresh of the name cache. This may be useful to ensure up-to-date name information is present in the cache after user and/or machine accounts are known to have changed and you need to see those changes immediately instead of waiting for the periodic automatic refresh.

The Broker Service maintains a cache of the names of users and machines in use by the site. By default, name information is obtained periodically from Active Directory and the cache refreshed automatically.

During normal usage, you should not need to perform administrative operations on the name cache.

For users/groups, the following name information is cached:

Windows name (DOMAIN\user)

User Principal Name or 'UPN' (user@upndomain)

Full Name or 'Common Name' (typically a user's full name)

For machines, the following name information is cached:

Windows name (DOMAIN\machine)

DNS name (machine.dnsdomain)

## Related topics

## Parameters

**-Machines**<SwitchParameter>

Triggers an asynchronous refresh of all cached machine name information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Users**<SwitchParameter>

Triggers an asynchronous refresh of all cached user name information.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snapin will connect to. This can be provided as a host name or an IP address.

|                        |                                                                                        |
|------------------------|----------------------------------------------------------------------------------------|
| Required?              | false                                                                                  |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value will become the default. |
| Accept Pipeline Input? | false                                                                                  |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

None

**Notes**

For some user accounts, for example, the built-in domain administrator, the UPN and/or Full Name values may not be available because they are not typically specified within Active Directory.

For group accounts, UPN and Full Name values are not available because they are not applicable or not specified within Active Directory.

The DNS name information for a machine is obtained from Active Directory and not from the DNS sub-system. If a machine has only recently been configured, the DNS information may not be available initially.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Update-BrokerNameCache -Machines

Triggers an immediate asynchronous refresh of all machine name information held within the name cache.

----- **EXAMPLE 2** -----

C:\PS> Update-BrokerNameCache -Machines -Users

Triggers an immediate asynchronous refresh of all machine and user name information held within the name cache.

# Citrix.Configuration.Admin.V2

Jan 04, 2017  
Overview

| Name                                      | Description                                                          |
|-------------------------------------------|----------------------------------------------------------------------|
| <a href="#">ConfigConfigurationSnapin</a> | The Configuration service PowerShell snap-in provides administrative |
| <a href="#">Config Filtering</a>          | Describes the common filtering options for XenDesktop cmdlets.       |

## Cmdlets

| Name                                                        | Description                                                                                        |
|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <a href="#">Add-ConfigRegisteredServiceInstanceMetadata</a> | Adds metadata on the given ServiceInstance.                                                        |
| <a href="#">Add-ConfigServiceGroupMetadata</a>              | Adds metadata on the given ServiceGroup.                                                           |
| <a href="#">Export-ConfigFeatureTable</a>                   | Returns the current feature table.                                                                 |
| <a href="#">Get-ConfigDBConnection</a>                      | Gets the database string for the specified data store used by the Configuration Service.           |
| <a href="#">Get-ConfigDBSchema</a>                          | Gets a script that creates the Configuration Service database schema for the specified data store. |
| <a href="#">Get-ConfigDBVersionChangeScript</a>             | Gets a script that updates the Configuration Service database schema.                              |
| <a href="#">Get-ConfigEnabledFeature</a>                    | Lists features of the site that are enabled.                                                       |
| <a href="#">Get-ConfigInstalledDBVersion</a>                | Gets a list of all available database schema versions for the Configuration Service.               |
| <a href="#">Get-ConfigLicensingModel</a>                    | Lists the supported licensing models.                                                              |
| <a href="#">Get-ConfigLocalData</a>                         | Gets the service local data.                                                                       |
| <a href="#">Get-ConfigProduct</a>                           | Lists the site's supported product names and codes.                                                |
| <a href="#">Get-ConfigProductEdition</a>                    | Lists the supported product editions.                                                              |

| Name                                           | Description                                                                                       |
|------------------------------------------------|---------------------------------------------------------------------------------------------------|
| Get-ConfigProductFeature                       | Lists the supported features.                                                                     |
| Get-ConfigProductVersion                       | Lists the supported product versions.                                                             |
| Get-ConfigRegisteredServiceInstance            | Gets the service instances that are registered in the directory.                                  |
| Get-ConfigService                              | Gets the service record entries for the Configuration Service.                                    |
| Get-ConfigServiceAddedCapability               | Gets any added capabilities for the Configuration Service on the controller.                      |
| Get-ConfigServiceGroup                         | Gets the service groups that match the parameters supplied.                                       |
| Get-ConfigServiceInstance                      | Gets the service instance entries for the Configuration Service.                                  |
| Get-ConfigServiceStatus                        | Gets the current status of the Configuration Service on the controller.                           |
| Get-ConfigSite                                 | Gets the site.                                                                                    |
| Import-ConfigFeatureTable                      | Sets the feature table of the site.                                                               |
| Register-ConfigServiceInstance                 | Allows the registration of a service instance.                                                    |
| Remove-ConfigRegisteredServiceInstanceMetadata | Removes metadata from the given ServiceInstance.                                                  |
| Remove-ConfigServiceGroup                      | Removes service groups.                                                                           |
| Remove-ConfigServiceGroupMetadata              | Removes metadata from the given ServiceGroup.                                                     |
| Remove-ConfigServiceMetadata                   | Removes metadata from the given Service.                                                          |
| Remove-ConfigSiteMetadata                      | Removes metadata from the given Site.                                                             |
| Reset-ConfigServiceGroupMembership             | Reloads the access permissions and configuration service locations for the Configuration Service. |
| Set-ConfigDBConnection                         | Configures a database connection for the Configuration Service.                                   |
| Set-ConfigRegisteredServiceInstance            | Updates a service instance.                                                                       |

| Name                                        | Description                                                              |
|---------------------------------------------|--------------------------------------------------------------------------|
| Set-ConfigRegisteredServiceInstanceMetadata | Adds or updates metadata on the given ServiceInstance.                   |
| Set-ConfigServiceGroupMetadata              | Adds or updates metadata on the given ServiceGroup.                      |
| Set-ConfigServiceMetadata                   | Adds or updates metadata on the given Service.                           |
| Set-ConfigSite                              | Changes the overall settings of the site.                                |
| Set-ConfigSiteMetadata                      | Adds or updates metadata on the Site.                                    |
| Test-ConfigDBConnection                     | Tests a database connection for the Configuration Service.               |
| Test-ConfigServiceInstanceAvailability      | Tests whether the supplied service instances are responding to requests. |
| Unregister-ConfigRegisteredServiceInstance  | Removes a service instance from the Configuration Service registry.      |

# about\_ConfigConfigurationSnapin

Jan 04, 2017

## TOPIC

about\_ConfigConfigurationSnapin

## SHORT DESCRIPTION

The Configuration service PowerShell snap-in provides administrative functions for the Configuration service.

## COMMAND PREFIX

All commands in this snap-in have 'Config' in their name.

## LONG DESCRIPTION

The Configuration service PowerShell snap-in enables both local and remote administration of the Configuration service. It provides facilities to store details about other services that are used in the XenDesktop deployment.

All the services in the XenDesktop deployment use the Configuration service as a directory to locate other services with which they need to communicate. The directory publishes a list of all the services, the communication types they accept, and the facilities they offer.

The snap-in provides the following main entities:

### Site Metadata

Metadata for the entire XenDesktop deployment. This metadata is not used directly by any of the XenDesktop services, but can be used by third parties to store information for other purposes.

### Registered Service Instances

Service instance items that have been retrieved from the available services in the XenDesktop deployment and held in a directory in the Configuration service. Each physical service can host a collection of service instances to provide different facilities.

### Service Groups

A collection of service instances that are considered equivalent. Service instances that are registered in the same service group provide the same state and offer the same functionality.

Only one service group of each type is supported. For example, there

must not be more than one service group with a ServiceType of 'Config'.

## Site and Features

The configuration site is a top-level, logical representation of the XenDesktop site, from the perspective of the configuration services running within the site.



# about\_Config\_Filtering

Jan 04, 2017

## TOPIC

XenDesktop - Advanced Dataset Filtering

## SHORT DESCRIPTION

Describes the common filtering options for XenDesktop cmdlets.

## LONG DESCRIPTION

Some cmdlets operate on large quantities of data and, to reduce the overhead of sending all of that data over the network, many of the Get- cmdlets support server-side filtering of the results.

The conventional way of filtering results in PowerShell is to pipeline them into Where-Object, Select-Object, and Sort-Object, for example:

```
Get-<Noun> | Where { $_.Size = 'Small' } | Sort 'Date' | Select -First 10
```

However, for most XenDesktop cmdlets the data is stored remotely and it would be slow and inefficient to retrieve large amounts of data over the network and then discard most of it. Instead, many of the Get- cmdlets provide filtering parameters that allow results to be processed on the server, returning only the required results.

You can filter results by most object properties using parameters derived from the property name. You can also sort results or limit them to a specified number of records:

```
Get-<Noun> -Size 'Small' -SortBy 'Date' -MaxRecordCount 10
```

You can express more complex filter conditions using a syntax and set of operators very similar to those used by PowerShell expressions.

Those cmdlets that support filtering have the following common parameters:

`-MaxRecordCount <int>`

Specifies the maximum number of results to return.  
For example, to return only the first nine results use:

```
Get-<Noun> -MaxRecordCount 9
```

If not specified, only the first 250 records are returned, and if more are available, a warning is produced:

WARNING: Only first 250 records returned. Use -MaxRecordCount to retrieve more.

You can suppress this warning by using -WarningAction or by specifying a value for -MaxRecordCount.

To retrieve all records, specify a large number for -MaxRecordCount. As the value is an integer, you can use the following:

```
Get-<Noun> -MaxRecordCount [int]::MaxValue
```

-ReturnTotalRecordCount [<SwitchParameter>]

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. For example:

```
Get-<Noun> -MaxRecordCount 9 -ReturnTotalRecordCount
```

....

```
Get-<Noun> : Returned 9 of 10 items
```

```
At line:1 char:18
```

```
+ Get-<Noun> <<<< -MaxRecordCount 9 -ReturnTotalRecordCount
```

```
+ CategoryInfo          : OperationStopped: (:) [Get-<Noun>], PartialDataException
```

```
+ FullyQualifiedErrorId : PartialData,Citrix.<SDKName>.SDK.Get<Noun>
```

The count can be accessed using the TotalAvailableResultCount property:

```
$count = $error[0].TotalAvailableResultCount
```

-Skip <int>

Skips the specified number of records before returning results.

Also reduces the count returned by -ReturnTotalRecordCount.

-SortBy <string>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order, respectively. Ascending order is assumed if no prefix is present.

Sorting occurs before -MaxRecordCount and -Skip parameters are applied. For example, to sort by Name and then by Count (largest first) use:

```
-SortBy 'Name,-Count'
```

By default, sorting by an enumeration property uses the numeric value of the elements. You can specify a different sort order by qualifying the name with an ordered list of elements or their numeric values, or <null> to indicate the placement of null values.

Elements not mentioned are placed at the end in their numeric order. For example, to sort by two different enums and then by the object id:

```
-SortBy 'MyState(StateC,<null>,StateA,StateB),Another(0,3,2,1),Id'
```

-Filter <String>

This parameter lets you specify advanced filter expressions, and supports combination of conditions with -and and -or, and grouping with braces. For example:

```
Get-<Noun> -Filter 'Name -like "High*" -or (Priority -eq 1 -and Severity -ge 2)'
```

The syntax is close enough to PowerShell syntax that you can use script blocks in most cases. This can be easier to read as it reduces quoting:

```
Get-<Noun> -Filter { Count -ne $null }
```

The full -Filter syntax is provided below.

## EXAMPLES

Filtering by strings performs a case-insensitive wildcard match.

Separate parameters are combined with an implicit -and operator.

Normal PowerShell quoting rules apply, so you can use single or double quotes, and omit the quotes altogether for many strings. The order of parameters does not make any difference. The following are equivalent:

```
Get-<Noun> -Company Citrix -Product Xen*
Get-<Noun> -Company "citrix" -Product '[X]EN*'
Get-<Noun> -Product "Xen*" -Company "CITRIX"
Get-<Noun> -Filter { Company -eq 'Citrix' -and Product -like 'Xen*' }
```

See [about\\_Quoting\\_Rules](#) and [about\\_Wildcards](#) for details about PowerShell handling of quotes and wildcards.

To avoid wildcard matching or include quote characters, you can escape the wildcards using the normal PowerShell escape mechanisms (see [about\\_Escape\\_Characters](#)), or switch to a filter expression and the `-eq` operator:

```
Get-<Noun> -Company "Abc[*]"           # Matches Abc*
Get-<Noun> -Company "Abc`*"           # Matches Abc*
Get-<Noun> -Filter { Company -eq "Abc*" } # Matches Abc*
Get-<Noun> -Filter { Company -eq "A`"B`C" } # Matches A"B'C
```

Simple filtering by numbers, booleans, and TimeSpans perform direct equality comparisons, although if the value is nullable you can also search for null values. Here are some examples:

```
Get-<Noun> -Uid 123
Get-<Noun> -Enabled $true
Get-<Noun> -Duration 1:30:40
Get-<Noun> -NullableProperty $null
```

More comparisons are possible using advanced filtering with `-Filter`:

```
Get-<Noun> -Filter 'Capacity -ge 10gb'
Get-<Noun> -Filter 'Age -ge 20 -and Age -lt 40'
Get-<Noun> -Filter 'VolumeLevel -like "[123]"'
Get-<Noun> -Filter 'Enabled -ne $false'
Get-<Noun> -Filter 'NullableProperty -ne $null'
```

You can check boolean values without an explicit comparison operator, and you can also combine them with `-not`:

```
Get-<Noun> -Filter 'Enabled' # Equivalent to 'Enabled -eq $true'
Get-<Noun> -Filter '-not Enabled' # Equivalent to 'Enabled -eq $false'
```

See [about\\_Comparison\\_Operators](#) for an explanation of the operators, but note that only a subset of PowerShell operators are supported (`-eq`, `-ne`, `-gt`, `-ge`, `-lt`, `-le`, `-like`, `-notlike`, `-in`, `-notin`, `-contains`, `-notcontains`).

Enumeration values can either be specified using typed values or the string name of the enumeration value:

```
Get-<Noun> -Shape [Shapes]::Square
Get-<Noun> -Shape Circle
```

With filter expressions, typed values can be specified with simple variables or quoted strings. They also support enumerations with wildcards:

```
$s = [Shapes]::Square
Get-<Noun> -Filter { Shape -eq $s -or Shape -eq "Circle" }
Get-<Noun> -Filter { Shape -like 'C*' }
```

By their nature, floating point values, DateTime values, and TimeSpan values are best suited to relative comparisons rather than just equality. DateTime strings are converted using the locale and time zone of the user device, but you can use ISO8601 format strings (YYYY-MM-DDThh:mm:ss.sTZD) to avoid ambiguity. You can also use standard PowerShell syntax to create these values:

```
Get-<Noun> -Filter { StartTime -ge "2010-08-23T12:30:00.0Z" }
$d = [DateTime]"2010-08-23T12:30:00.0Z"
Get-<Noun> -Filter { StartTime -ge $d }
$d = (Get-Date).AddDays(-1)
Get-<Noun> -Filter { StartTime -ge $d }
```

Relative times are quite common and, when using filter expressions, you can also specify DateTime values using a relative format:

```
Get-<Noun> -Filter { StartTime -ge '-2' } # Two days ago
Get-<Noun> -Filter { StartTime -ge '-1:30' } # Hour and a half ago
Get-<Noun> -Filter { StartTime -ge '-0:0:30' } # 30 seconds ago
```

## ARRAY PROPERTIES

When filtering against list or array properties, simple parameters perform a case-insensitive wildcard match against each of the members. With filter expressions, you can use the -contains and -notcontains operators. Unlike PowerShell, these perform wildcard matching on strings.

Note that for array properties the naming convention is for the returned property to be plural, but the parameter used to search for any match is singular. The following are equivalent (assuming Users is an array property):

```
Get-<Noun> -User Fred*
Get-<Noun> -Filter { User -like "Fred*" }
Get-<Noun> -Filter { Users -contains "Fred*" }
```

You can also use the singular form with -Filter to search using other operators:

```
# Match if any user in the list is called "Frederick"
Get-<Noun> -Filter { User -eq "Frederick" }
# Match if any user in the list has a name alphabetically below 'F'
Get-<Noun> -Filter { User -lt 'F' }
```

## COMPLEX EXPRESSIONS

When matching against multiple values, you can use a sequence of comparisons joined with -or operators, or you can use -in and -notin:

```
Get-<Noun> -Filter { Shape -eq 'Circle' -or Shape -eq 'Square' }
$shapes = 'Circle','Square'
Get-<Noun> -Filter { Shape -in $shapes }
$sides = 1..4
Get-<Noun> -Filter { Sides -notin $sides }
```

Braces can be used to group complex expressions, and override the default left-to-right evaluation of -and and -or. You can also use -not to invert the sense of any sub-expression:

```
Get-<Noun> -Filter { Size -gt 4 -or (Color -eq 'Blue' -and Shape -eq 'Circle') }
Get-<Noun> -Filter { Sides -lt 5 -and -not (Color -eq 'Blue' -and Shape -eq 'Circle') }
```

## PAGING

The simplest way to page through data is to use the -Skip and -MaxRecordCount parameters. So, to read the first three pages of data with 10 records per page, use:

```
Get-<Noun> -Skip 0 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 10 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 20 -MaxRecordCount 10 <other filtering criteria>
```

You must include the same filtering criteria on each call, and ensure that the data is sorted consistently.

The above approach is often acceptable, but as each call performs an independent query, data changes can result in records being skipped or appearing twice. One approach to improve this is to sort by a unique id field and then start the search for the next page at the unique id after the last unique id of the previous page. For example:

```
# Get the first page
Get-<Noun> -MaxRecordCount 10 -SortBy SerialNumber

SerialNumber ...
----- ---
A120004
A120007
... 7 other records ...
A120900

# Get the next page
Get-<Noun> -MaxRecordCount 10 -Filter { FirstName -gt 'A120900' }

SerialNumber ...
```

----- ---  
A120901  
B220000  
...

## FILTER SYNTAX DEFINITION

<Filter> ::= <ScriptBlock> | <ComponentList>

<ScriptBlock> ::= "{" <ComponentList> "}"

<ComponentList> ::= <Component> <AndOrOperator> <ComponentList> |

<Component>

<Component> ::= <NotOperator> <Factor> |

<Factor>

<Factor> ::= "(" <ComponentList> ")" |

<PropertyName> <ComparisonOperator> <Value> |  
<PropertyName>

<AndOrOperator> ::= "-and" | "-or"

<NotOperator> ::= "-not" | "!"

<ComparisonOperator>

::= "-eq" | "-ne" | "-le" | "-ge" | "-lt" | "-gt" |  
"-like" | "-notlike" | "-contains" | "-notcontains" |  
"-in" | "-notin"

<PropertyName> ::= <simple name of property>

<Value> ::= <string literal> | <numeric literal> |

<scalar variable> | <array variable> |  
"\$null" | "\$true" | "\$false"

Numeric literals support decimal and hexadecimal literals, with optional multiplier suffixes (kb, mb, gb, tb, pb).

Dates and times can be specified as string literals. The current culture determines what formats are accepted. To avoid any ambiguity, use strings formatted to the ISO8601 standard. If not specified, the current time zone is used.

Relative date-time string literals are also supported, using a minus sign followed by a TimeSpan. For example, "-1:30" means 1 hour and 30 minutes ago.



# Add-ConfigRegisteredServiceInstanceMetadata

Jan 04, 2017

Adds metadata on the given ServiceInstance.

## Syntax

```
Add-ConfigRegisteredServiceInstanceMetadata [-ServiceInstanceId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ConfigRegisteredServiceInstanceMetadata [-ServiceInstanceId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ConfigRegisteredServiceInstanceMetadata [-InputObject] <ServiceInstance[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ConfigRegisteredServiceInstanceMetadata [-InputObject] <ServiceInstance[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given ServiceInstance objects. This cmdlet will not overwrite existing metadata on an object - use the Set-ConfigRegisteredServiceInstanceMetadata cmdlet instead.

## Related topics

[Set-ConfigRegisteredServiceInstanceMetadata](#)

[Remove-ConfigRegisteredServiceInstanceMetadata](#)

## Parameters

**-ServiceInstanceId**<Guid>

Id of the ServiceInstance

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<ServiceInstance[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the ServiceInstance specified. The property cannot contain any of the following characters \/:#.\*?=<>|[]0''

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Return Values

Citrix.Configuration.Sdk.Metadata

Add-ConfigRegisteredServiceInstanceMetadata returns an array of objects containing the new definition of the metadata.

\n Property <string>

\n Specifies the name of the property.

\n Value <string>

\n Specifies the value for the property.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DuplicateObject

One of the specified metadata already exists.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Add-ConfigRegisteredServiceInstanceMetadata -ServiceInstanceUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Property

Value

-----

property

-----

value

Add metadata with a name of 'property' and a value of 'value' to the ServiceInstance with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Add-ConfigServiceGroupMetadata

Jan 04, 2017

Adds metadata on the given ServiceGroup.

## Syntax

```
Add-ConfigServiceGroupMetadata [-ServiceGroupUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ConfigServiceGroupMetadata [-ServiceGroupUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ConfigServiceGroupMetadata [-InputObject] <ServiceGroup[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ConfigServiceGroupMetadata [-InputObject] <ServiceGroup[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given ServiceGroup objects. This cmdlet will not overwrite existing metadata on an object - use the Set-ConfigServiceGroupMetadata cmdlet instead.

## Related topics

[Set-ConfigServiceGroupMetadata](#)

[Remove-ConfigServiceGroupMetadata](#)

## Parameters

**-ServiceGroupUid**<Guid>

Id of the ServiceGroup

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<ServiceGroup[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Map<PSObject>**

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name<String>**

Specifies the property name of the metadata to be added. The property must be unique for the ServiceGroup specified. The property cannot contain any of the following characters \/:#.\*?=<>|[]()"

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value<String>**

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId<Guid>**

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP

address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.Configuration.Sdk.Metadata

Add-ConfigServiceGroupMetadata returns an array of objects containing the new definition of the metadata.

\n Property <string>

\n Specifies the name of the property.

\n Value <string>

\n Specifies the value for the property.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DuplicateObject

One of the specified metadata already exists.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Add-ConfigServiceGroupMetadata -ServiceGroupUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Property | Value |
|----------|-------|
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the ServiceGroup with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.



# Export-ConfigFeatureTable

Jan 04, 2017

Returns the current feature table.

## Syntax

```
Export-ConfigFeatureTable [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

## Related topics

[Set-ConfigSite](#)

[Import-ConfigFeatureTable](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.String

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Export-ConfigFeatureTable
<?xml version="1.0" encoding="utf-8">
<Products xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="FeatureTable.xsd">
  <Product Code="XDT" Name="XenDesktop">
    <Editions>
      <Edition>PLT</Edition>
      <Edition>ENT</Edition>
      <Edition>APP</Edition>
      <Edition>STD</Edition>
    </Editions>
    <DefaultEdition>PLT</DefaultEdition>
    <VersionToBurninDates>
      <VersionToBurninDate Version="7.0" BurninDate="2013.0522" />
    </VersionToBurninDates>
    <LicensingModels>
      <LicensingModel>Concurrent</LicensingModel>
      <LicensingModel>UserDevice</LicensingModel>
    </LicensingModels>
    <DefaultLicensingModel>UserDevice</DefaultLicensingModel>
    <Features>
      <Feature Name="CustomRole">
```

```

<BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="ENT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="APP" MustBe="AtLeast">2013.0522</BurninDate>
</Feature>
<Feature Name="ConfigurationLogging">
<BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="ENT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="APP" MustBe="AtLeast">2013.0522</BurninDate>
</Feature>
<Feature Name="SingleUserMode">
<BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="ENT" MustBe="AtLeast">2013.0522</BurninDate>
</Feature>
<Feature Name="MultiSessionDesktops">
<BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="ENT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="APP" MustBe="AtLeast">2013.0522</BurninDate>
</Feature>
<Feature Name="SingleSessionDesktops">
<BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="ENT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="STD" MustBe="AtLeast">2013.0522</BurninDate>
</Feature>
<Feature Name="MultiSessionApplications">
<BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="ENT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="APP" MustBe="AtLeast">2013.0522</BurninDate>
</Feature>
<Feature Name="SingleSessionApplications">
<BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="ENT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="APP" MustBe="AtLeast">2013.0522</BurninDate>
</Feature>
<Feature Name="HistoricalMonitorData">
<BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
</Feature>
<Feature Name="CloudHostedMachines">
<BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
</Feature>
<Feature Name="HdxInsightIntegration">
<BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
</Feature>
<Feature Name="RemotePC">
<BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
<BurninDate ForProductEdition="ENT" MustBe="AtLeast">2013.0522</BurninDate>
</Feature>
</Features>
</Product>
<Product Code="MPS" Name="XenApp">
<Editions>
<Edition>PLT</Edition>
<Edition>ENT</Edition>
</Editions>
<DefaultEdition>PLT</DefaultEdition>
<VersionToBurninDates>
<VersionToBurninDate Version="7.0" BurninDate="2013.0522" />
</VersionToBurninDates>

```

```
<LicensingModels>
  <LicensingModel>Concurrent</LicensingModel>
</LicensingModels>
<DefaultLicensingModel>Concurrent</DefaultLicensingModel>
<Features>
  <Feature Name="CustomRole">
    <BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
    <BurninDate ForProductEdition="ENT" MustBe="AtLeast">2013.0522</BurninDate>
  </Feature>
  <Feature Name="ConfigurationLogging">
    <BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
    <BurninDate ForProductEdition="ENT" MustBe="AtLeast">2013.0522</BurninDate>
  </Feature>
  <Feature Name="SingleUserMode">
  </Feature>
  <Feature Name="MultiSessionDesktops">
    <BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
    <BurninDate ForProductEdition="ENT" MustBe="AtLeast">2013.0522</BurninDate>
  </Feature>
  <Feature Name="SingleSessionDesktops">
  </Feature>
  <Feature Name="MultiSessionApplications">
    <BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
    <BurninDate ForProductEdition="ENT" MustBe="AtLeast">2013.0522</BurninDate>
  </Feature>
  <Feature Name="SingleSessionApplications">
    <BurninDate ForProductEdition="PLT" MustBe="AtLeast">2013.0522</BurninDate>
    <BurninDate ForProductEdition="ENT" MustBe="AtLeast">2013.0522</BurninDate>
  </Feature>
  <Feature Name="HistoricalMonitorData">
  </Feature>
  <Feature Name="CloudHostedMachines">
  </Feature>
  <Feature Name="HdxInsightIntegration">
  </Feature>
  <Feature Name="RemotePC">
  </Feature>
</Features>
</Product>
</Products>
```

Returns the current feature table.

# Get-ConfigDBConnection

Jan 04, 2017

Gets the database string for the specified data store used by the Configuration Service.

## Syntax

```
Get-ConfigDBConnection [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the database connection string for the specified data store.

If the returned string is blank, no valid connection string has been specified. In this case the service is running, but is idle and awaiting specification of a valid connection string.

## Related topics

[Get-ConfigServiceStatus](#)

[Set-ConfigDBConnection](#)

[Test-ConfigDBConnection](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

system.string

The database connection string configured for the Configuration Service.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

NoDBConnections

The database connection string for the Configuration Service has not been specified.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigDBConnection
```

Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True  
Get the database connection string for the Configuration Service.

# Get-ConfigDBSchema

Jan 04, 2017

Gets a script that creates the Configuration Service database schema for the specified data store.

## Syntax

```
Get-ConfigDBSchema [-DatabaseName <String>] [-ServiceGroupName <String>] [-ScriptType <ScriptTypes>] [-LocalDatabase] [-Sid <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets SQL scripts that can be used to create a new Configuration Service database schema, add a new Configuration Service to an existing site, remove a Configuration Service from a site, or create a database server logon for a Configuration Service. If no Sid parameter is provided, the scripts obtained relate to the currently selected Configuration Service instance, otherwise the scripts relate to Configuration Service instance running on the machine identified by the Sid provided. When obtaining the Evict script, a Sid parameter must be supplied. The current service instance is that on the local machine, or that explicitly specified by the last usage of the -AdminAddress parameter to a Configuration SDK cmdlet. The service instance used to obtain the scripts does not need to be a member of a site or to have had its database connection configured. The database scripts support only Microsoft SQL Server, or SQL Server Express, and require Windows integrated authentication to be used. They can be run using SQL Server's SQLCMD utility, or by copying the script into an SQL Server Management Studio (SSMS) query window and executing the query. If using SSMS, the query must be executed in 'SMDCMD mode'. The ScriptType parameter determines which script is obtained. If ScriptType is not specified, or is FullDatabase, the script contains:

- o Creation of service schema
- o Creation of database server logon
- o Creation of database user
- o Addition of database user to Configuration Service roles

If ScriptType is Instance, the returned script contains:

- o Creation of database server logon
- o Creation of database user
- o Addition of database user to Configuration Service roles

If ScriptType is Evict, the returned script contains:

- o Removal of Configuration Service instance from database
- o Removal of database user

If ScriptType is Login, the returned script contains:

- o Creation of database server logon only

If the service uses two data stores they can exist in the same database. You do not need to configure a database before

using this command.

## Related topics

[Set-ConfigDBConnection](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database for which the schema will be generated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ServiceGroupName**<String>

Specifies the name of the service group to be used when creating the database schema. The service group is a collection of all the Configuration services that share the same database instance and are considered equivalent; that is, all the services within a service group can be used interchangeably.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Script Type**<ScriptTypes>

Specifies the type of database script returned. Available script types are:

### Database

Returns a full database script that can be used to create a database schema for the Configuration Service in a database instance that does not already contain a schema for this service. The DatabaseName and ServiceGroupName parameters must be specified to create a script of this type.

### Instance

Returns a permissions script that can be used to add further Configuration services to an existing database instance that already contains the full Configuration service schema, associating the services to the Service Group. The Sid parameter can optionally be specified to create a script of this type.

### Login

Returns a database logon script that can be used to add the required logon accounts to an existing database instance that contains the Configuration Service schema. This is used primarily when creating a mirrored database environment. The DatabaseName parameter must be specified to create a script of this type.

**Evict**

Returns a script that can be used to remove the specified Configuration Service from the database entirely. The DatabaseName and Sid parameters must be specified to create a script of this type.

|                        |          |
|------------------------|----------|
| Required?              | false    |
| Default Value          | Database |
| Accept Pipeline Input? | false    |

**-LocalDatabase**<SwitchParameter>

Specifies whether the database script is to be used in a database instance run on the same controller as other services in the service group. Including this parameter ensures the script creates only the required permissions for local services to access the database schema for Configuration services.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Sid**<String>

Specifies the SID of the controller on which the Configuration Service instance to remove from the database is running.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

### Systemstring

A string containing the required SQL script for application to a database.

## Notes

The scripts returned support Microsoft SQL Server Express Edition, Microsoft SQL Server Standard Edition, and Microsoft SQL Server Enterprise Edition databases only, and are generated on the assumption that integrated authentication will be used.

If the ScriptType parameter is not included or set to 'FullDatabase', the full database script is returned, which will:

Create the database schema.

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist).

If the ScriptType parameter is set to 'Instance', the script will:

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist) and associate it with a user.

If the ScriptType parameter is set to 'Login', the script will:

Create the logon (providing the schema does not already exist) and associate it with a pre-existing user of the same name.

If the LocalDatabase parameter is included, the NetworkService account will be added to the list of accounts permitted to access the database. This is required only if the database is run on a controller.

If the command fails, the following errors can be returned.

### Error Codes

-----

#### GetSchemasFailed

The database schema could not be found.

#### ActiveDirectoryAccountResolutionFailed

The specified Active Directory account or Group could not be found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigDBSchema -DatabaseName MyDB -ServiceGroupName MyServiceGroup > c:\ConfigSchema.sql  
Get the full database schema for site data store of the Configuration Service and copy it to a file called  
'c:\ConfigSchema.sql'.
```

This script can then be used to create the schema in a pre-existing database named 'MyDB' that does not already contain a Configuration Service site schema.

----- **EXAMPLE 2** -----

```
c:\PS>Get-ConfigDBSchema -DatabaseName MyDB -scriptType Login > c:\ConfigurationLogins.sql  
Get the logon scripts for the Configuration Service.
```

# Get-ConfigDBVersionChangeScript

Jan 04, 2017

Gets a script that updates the Configuration Service database schema.

## Syntax

```
Get-ConfigDBVersionChangeScript -DatabaseName <String> -TargetVersion <Version> [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Returns a database script that can be used to upgrade or downgrade the site or secondary schema for the Configuration Service from the current schema version to a different version.

## Related topics

[Get-ConfigInstalledDBVersion](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database instance to which the update applies.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TargetVersion**<Version>

Specifies the version of the database you want to update to.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.Management.Automation.PSObject

A PSObject containing the required SQL script for application to a database.

## Notes

The PSObject returned by this cmdlet contains the following properties:

- Script The raw text of the SQL script to apply the update, or null in the case when no upgrade path to the specified target version exists.
- NeedExclusiveAccess Indicates whether all services in the service group must be shut down during the update or not.
- CanUndo Indicates whether the generated script allows the updated schema to be reverted to the state prior to the update.

Scripts to update the schema version are stored in the database so any service in the service group can obtain these scripts. Extreme caution should be exercised when using update scripts. Citrix recommends backing up the database before attempting to upgrade the schema. Database update scripts may require exclusive use of the schema and so may not be able to execute while any Configuration services are running. However, this depends on the specific update being carried out.

After a schema update has been carried out, services that require the previous version of the schema may cease to operate. The ServiceState parameter reported by the Get-ConfigServiceStatus command provides information about service compatibility. For example, if the schema has been upgraded to a more recent version that a service cannot use, the service reports "DBNewerVersionThanService".

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoOp

The operation was successful but had no effect.

#### NoDBConnections

The database connection string for the Configuration Service has not been specified.

#### DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $update = Get-ConfigDBVersionChangeScript -DatabaseName MyDb -TargetVersion 1.0.75.0
```

```
C:\PS> $update.Script > update_75.sql
```

Gets an SQL update script to update the current schema to version 1.0.75.0. The resulting update\_75.sql script is suitable for direct use with the SQL Server SQLCMD utility.

# Get-ConfigEnabledFeature

Jan 04, 2017

Lists features of the site that are enabled.

## Syntax

```
Get-ConfigEnabledFeature [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

## Related topics

[Set-ConfigSite](#)

[Import-ConfigFeatureTable](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.String

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-ConfigEnabledFeature
```

Retrieves the list of enabled features for the site's current configuration.

# Get-ConfigInstalledDBVersion

Jan 04, 2017

Gets a list of all available database schema versions for the Configuration Service.

## Syntax

```
Get-ConfigInstalledDBVersion [-Upgrade] [-Downgrade] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the current version of the Configuration Service database schema, if no flags are set, otherwise returns versions for which upgrade or downgrade scripts are available and have been stored in the database.

## Related topics

## Parameters

**-Upgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be updated should be returned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Downgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be reverted should be returned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

### System.Version

The Get-ConfigInstalledDbVersion command returns objects containing the new definition of the Configuration Service database schema version.

Major <Integer>

Minor <Integer>

Build <Integer>

Revision <Integer>

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

Both the Upgrade and Downgrade flags were specified.

#### NoOp

The operation was successful but had no effect.

#### NoDBConnections

The database connection string for the Configuration Service has not been specified.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.



AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-ConfigInstalledDBVersion
```

```
Major Minor Build Revision
```

```
-----
```

```
5 6 0 0
```

Get the currently installed version of the Configuration Service database schema.

### ----- EXAMPLE 2 -----

```
c:\PS>Get-ConfigInstalledDBVersion -Upgrade
```

```
Major Minor Build Revision
```

```
-----
```

```
6 0 0 0
```

Get the versions of the Configuration Service database schema for which upgrade scripts are supplied.

# Get-ConfigLicensingModel

Jan 04, 2017

Lists the supported licensing models.

## Syntax

```
Get-ConfigLicensingModel -ProductCode <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

## Related topics

[Get-ConfigProduct](#)

[Get-ConfigSite](#)

[Set-ConfigSite](#)

[Import-ConfigFeatureTable](#)

## Parameters

**-ProductCode**<String>

The product code

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.String

The list of supported licensing models for the specified product code.

## Notes

The Get-ConfigProduct cmdlet lists the available product codes.

The site object returned by the Get-ConfigSite cmdlet contains the currently configured product code.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-ConfigLicensingModel -ProductCode "XDS"
```

Retrieves the list of supported licensing models for product code "XDS".

# Get-ConfigLocalData

Jan 04, 2017

Gets the service local data.

## Syntax

```
Get-ConfigLocalData [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Get-ConfigLocalData returns service-and-controller-specific local data. This information is not site-wide, but rather controller-specific. The Configuration Service currently stores the controller product version as local data. The overall site version used by Studio is an aggregate of the product versions from each controller.

## Related topics

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Input Type

This cmdlet does not accept pipeline input

## Return Values

Citrix.Configuration.DataModel.LocalData

Contains the ControllerProductVersion field

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-ConfigLocalData
```

```
ControllerProductVersion
```

```
-----
```

```
7.1
```

Gets the service local data.



# Get-ConfigProduct

Jan 04, 2017

Lists the site's supported product names and codes.

## Syntax

```
Get-ConfigProduct [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

## Related topics

[Get-ConfigSite](#)

[Set-ConfigSite](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

PObject

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-ConfigProduct
```

Lists the supported products by name and code.

# Get-ConfigProductEdition

Jan 04, 2017

Lists the supported product editions.

## Syntax

```
Get-ConfigProductEdition [-ProductCode] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

## Related topics

[Get-ConfigProduct](#)

[Get-ConfigSite](#)

[Set-ConfigSite](#)

[Import-ConfigFeatureTable](#)

## Parameters

**-ProductCode**<String>

The product code

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.String

The list of supported licensing models for the specified product code.

## Notes

The Get-ConfigProduct cmdlet lists the available product codes.

The site object returned by Get-ConfigSite cmdlet contains the currently configured product code.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-ConfigProductEdition -ProductCode "XDS"  
Retrieves the list of supported editions for XenDesktop.
```



# Get-ConfigProductFeature

Jan 04, 2017

Lists the supported features.

## Syntax

```
Get-ConfigProductFeature [-ProductCode] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Lists the supported features. Use the `Get-ConfigEnabledFeature` command to determine which features are currently enabled.

## Related topics

[Get-ConfigProduct](#)

[Get-ConfigSite](#)

[Set-ConfigSite](#)

[Import-ConfigFeatureTable](#)

## Parameters

**-ProductCode**<String>

The product code

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.String

The list of supported licensing models for the specified product code.

## Notes

The Get-ConfigProduct cmdlet lists the available product codes.

The site object returned by Get-ConfigSite cmdlet contains the currently configured product code.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-ConfigProductFeature -ProductCode "XDS"
```

Retrieves the list of supported features for XenDesktop.

# Get-ConfigProductVersion

Jan 04, 2017

Lists the supported product versions.

## Syntax

```
Get-ConfigProductVersion [-ProductCode] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

## Related topics

[Get-ConfigProduct](#)

[Get-ConfigSite](#)

[Set-ConfigSite](#)

[Import-ConfigFeatureTable](#)

## Parameters

**-ProductCode**<String>

The product code

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.String

The list of supported licensing models for the specified product code.

## Notes

The Get-ConfigProduct cmdlet lists the available product codes.

The site object returned by Get-ConfigSite cmdlet contains the currently configured product code.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-ConfigProductVersion -ProductCode "XDS"  
Retrieves the list of supported versions for XenDesktop.
```

# Get-ConfigRegisteredServiceInstance

Jan 04, 2017

Gets the service instances that are registered in the directory.

## Syntax

```
Get-ConfigRegisteredServiceInstance [-ServiceInstanceUid <Guid>] [-ServiceGroupUid <Guid>] [-ServiceGroupName <String>] [-ServiceType <String>] [-Address <String>] [-Binding <String>] [-Version <Int32>] [-ServiceAccountSid <String>] [-InterfaceType <String>] [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this cmdlet to retrieve the service instances currently registered with the Configuration Service that match the parameters supplied. If no parameters are supplied, all the service instances are returned.

## Related topics

[Register-ConfigServiceInstance](#)

[Unregister-ConfigRegisteredServiceInstance](#)

[Set-ConfigRegisteredServiceInstance](#)

## Parameters

**-ServiceInstanceUid**<Guid>

The unique identifier for the service instance.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ServiceGroupUid**<Guid>

The unique identifier for the service group to which the service instance belongs.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ServiceGroupName**<String>

The name for the service group to which the service instance belongs.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ServiceType**<String>

The service type for the service instance.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Address**<String>

The connection address for the service instance.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Binding**<String>

The binding for the service instance.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Version**<Int32>

The service instance version.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ServiceAccountSid**<String>

The AD account SID for the computer account that the computer hosting the service instance is running as.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InterfaceType**<String>

The interface type for the service instance.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

See about\_Config\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

See about\_Config\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

See about\_Config\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |



**-SortBy**<String>

See about\_Config\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Filter**<String>

See about\_Config\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | LocalHost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

Citrix.Configuration.Sdk.ServiceInstance

This represents a service instance and has the following parameters;

ServiceGroupUid <Guid>

The unique identifier for the service group to which the service instance belongs.

ServiceGroupName <string>

The name of the service group to which the service instance belongs.

ServiceInstanceUid <Guid>

The unique identifier for the service instance.

ServiceType <string>

The type of the service group.

Address <string>

The contact address for the service instance.

Binding <string>

The binding to use for connections to the service instance.

Version <int>

The version of the service instance.

ServiceAccount <string>

The AD computer account for the computer that is providing the service instance.

ServiceAccountSid <string>

The AD computer account SID for the computer that is providing the service instance.

InterfaceType <string>

The interface type for the service instance.

Metadata <Citrix.Configuration.Sdk.Metadata[]>

The metadata for the service instance.

## Notes

In the case of failure, the following errors can result.

**Error Codes** ----- PartialData Only a subset of the available data was returned. CouldNotQueryDatabase The query required to get the database was not defined. CommunicationError An error occurred while communicating with the service. DatabaseNotConfigured The operation could not be completed because the database for the service is not configured. InvalidFilter A filtering expression was supplied that could not be interpreted for this cmdlet. ExceptionThrown An unexpected error occurred. To locate more details see the Windows event logs on the controller being used, or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
C:\>Get-ConfigRegisteredServiceInstance
```

```
Address      : http://MyServer.com:80/Citrix/MyContract/v1
Binding      : wcf_HTTP_kerb
```

InterfaceType : SDK  
Metadata : {}  
MetadataMap : {}  
ServiceAccount : ENG\MyAccount\$\br/>ServiceAccountSid : S-1-5-21-1155438255-2213498043-2452000591-1104  
ServiceGroupName : MyService  
ServiceGroupUid : 2b990d5a-bba9-413b-aa08-e104e67f89bc  
ServiceInstanceUid : 8dc38b5a-3fbb-457c-b326-6c41c94c18d5  
ServiceType : MySnapIn  
Version : 1

Address : http://MyServer.com:80/Citrix/MyContract/PeerAPI/v1  
Binding : wcf\_HTTP\_kerb  
InterfaceType : Peer  
Metadata : {}  
MetadataMap : {}  
ServiceAccount : ENG\MyAccount\$\br/>ServiceAccountSid : S-1-5-21-1155438255-2213498043-2452000591-1104  
ServiceGroupName : MyService  
ServiceGroupUid : 2b990d5a-bba9-413b-aa08-e104e67f89bc  
ServiceInstanceUid : 8f822ed6-42f3-4a26-911a-a4a6a87c0ef2  
ServiceType : MySnapIn  
Version : 1

Address : http://MyServer.com:80/Citrix/MyContract/MyServiceEnvTestAPI/v1  
Binding : wcf\_HTTP\_kerb  
InterfaceType : EnvironmentTest  
Metadata : {}  
MetadataMap : {}  
ServiceAccount : ENG\MyAccount\$\br/>ServiceAccountSid : S-1-5-21-1155438255-2213498043-2452000591-1104  
ServiceGroupName : MyService  
ServiceGroupUid : 2b990d5a-bba9-413b-aa08-e104e67f89bc  
ServiceInstanceUid : d2d40d9b-2a5d-4c5a-b9ca-a7f73cffe4f2  
ServiceType : MySnapIn  
Version : 1

Address : http://MyServer.com:80/Citrix/MyContract/MyServiceAPI/v1  
Binding : wcf\_HTTP\_kerb  
InterfaceType : InterService  
Metadata : {}  
MetadataMap : {}  
ServiceAccount : ENG\MyAccount\$\br/>ServiceAccountSid : S-1-5-21-1155438255-2213498043-2452000591-1104  
ServiceGroupName : MyService  
ServiceGroupUid : 2b990d5a-bba9-413b-aa08-e104e67f89bc  
ServiceInstanceUid : 5d428970-2ba1-4336-b8d0-f3aa961b8983  
ServiceType : MySnapIn

Version : 1

Return all the service instances that are registered in the Configuration Service.

----- **EXAMPLE 2** -----

```
C:\>Get-ConfigRegisteredServiceInstance -InterfaceType "SDK"
```

Address : http://MyServer.com:80/Citrix/MyContract/v1

Binding : wcf\_HTTP\_kerb

InterfaceType : SDK

Metadata : {}

MetadataMap : {}

ServiceAccount : ENG\MyAccount\$

ServiceAccountSid : S-1-5-21-1155438255-2213498043-2452000591-1104

ServiceGroupName : MyService

ServiceGroupUid : 2b990d5a-bba9-413b-aa08-e104e67f89bc

ServiceInstanceUid : 8dc38b5a-3fbb-457c-b326-6c41c94c18d5

ServiceType : MySnapIn

Version : 1

Return all the service instances that are registered in the Configuration Service and are of type 'SDK'.

# Get-ConfigService

Jan 04, 2017

Gets the service record entries for the Configuration Service.

## Syntax

```
Get-ConfigService [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns instances of the Configuration Service that the service publishes. The service records contain account security identifier information that can be used to remove each service from the database.

A database connection for the service is required to use this command.

## Related topics

### Parameters

#### **-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: `-Metadata "abc:x"` matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See `about_Config_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by `-ReturnTotalRecordCount`.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

### **-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Config\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

Citrix.Configuration.Sdk.Service

The Get-ConfigServiceInstance command returns an object containing the following properties.

Uid <Integer>

Specifies the unique identifier for the service in the group. The unique identifier is an index number.

ServiceHostId <Guid>

Specifies the unique identifier for the service instance.

DNSName <String>

Specifies the domain name of the host on which the service runs.

MachineName <String>

Specifies the short name of the host on which the service runs.

CurrentState <Citrix.Fma.Sdk.ServiceCore.ServiceState>

Specifies whether the service is running, started but inactive, stopped, or failed.

LastStartTime <DateTime>

Specifies the date and time at which the service was last restarted.

LastActivityTime <DateTime>

Specifies the date and time at which the service was last stopped or restarted.

OSType

Specifies the operating system installed on the host on which the service runs.

OSVersion

Specifies the version of the operating system installed on the host on which the service runs.

ServiceVersion

Specifies the version number of the service instance. The version number is a string that reflects the full build version of the service.

DatabaseUserName <string>

Specifies for the service instance the Active Directory account name with permissions to access the database. This will be either the machine account or, if the database is running on a controller, the NetworkService account.

Sid <string>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

ActiveSiteServices <string[]>

Specifies the names of active site services currently running in the service. Site services are components that perform long-running background processing in some services. This field is empty for services that do not contain site services.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

UnknownObject

One of the specified objects was not found.

PartialData

Only a subset of the available data was returned.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

CouldNotQueryDatabase



The query required to get the database was not defined.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigService
```

```
Uid          : 1
ServiceHostId : aef6f464-f1ee-4042-a523-66982e0cecd0
DNSName      : MyServer.company.com
MachineName  : MYSERVER
CurrentState  : On
LastStartTime : 04/04/2011 15:25:38
LastActivityTime : 04/04/2011 15:33:39
OSType       : Win32NT
OSVersion    : 6.1.7600.0
ServiceVersion : 5.1.0.0
DatabaseUserName : NT AUTHORITY\NETWORK SERVICE
SID          : S-1-5-21-2316621082-1546847349-2782505528-1165
ActiveSiteServices : {MySiteService1, MySiteService2...}
Get all the instances of the Configuration Service running in the current service group.
```

# Get-ConfigServiceAddedCapability

Jan 04, 2017

Gets any added capabilities for the Configuration Service on the controller.

## Syntax

```
Get-ConfigServiceAddedCapability [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables updates to the Configuration Service on the controller to be detected.

You do not need to configure a database connection before using this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.String

String containing added capabilities.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigServiceAddedCapability
```

Get the added capabilities of the Configuration Service.

# Get-ConfigServiceGroup

Jan 04, 2017

Gets the service groups that match the parameters supplied.

## Syntax

```
Get-ConfigServiceGroup [-ServiceGroupUid <Guid>] [-ServiceGroupName <String>] [-ServiceType <String>] [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this cmdlet to retrieve existing service groups that match the parameters supplied. If no parameters are supplied, all the service groups are returned.

## Related topics

## Parameters

### **-ServiceGroupUid**<Guid>

The unique identifier for the service group.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-ServiceGroupName**<String>

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-ServiceType**<String>

The service type for the service group.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

See about\_Config\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

See about\_Config\_Filtering for details.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

See about\_Config\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

See about\_Config\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Filter**<String>

See about\_Config\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|           |       |

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | LocalHost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.Configuration.Sdk.ServiceGroup

This represents a service instance and has the following parameters;

ServiceGroupUid <Guid>

The unique identifier for the service group.

ServiceGroupName <string>

The name of the service group.

ServiceType <string>

The type of the service group.

Metadata <Citrix.Configuration.Sdk.Metadata[]>

The metadata for the service group.

## Notes

In the case of failure, the following errors can result.

Error Codes -----

DatabaseNotConfigured The operation could not be completed because the database for the service is not configured.  
 PartialData Only a subset of the available data was returned. CouldNotQueryDatabase The Query required to get the database was not defined. CommunicationError An error occurred while communicating with the service. InvalidFilter A filtering expression was supplied that could not be interpreted for this cmdlet. ExceptionThrown An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\>Get-ConfigServiceGroup
```

Return all the service groups that are registered in the Configuration Service.

### ----- EXAMPLE 2 -----

```
C:\>Get-ConfigRegisteredServiceInstance -ServiceType "config"
```

Return all the service groups that are registered in the Configuration Service and are of type 'config' (i.e. the service groups for the Configuration Service).

# Get-ConfigServiceInstance

Jan 04, 2017

Gets the service instance entries for the Configuration Service.

## Syntax

```
Get-ConfigServiceInstance [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns service interfaces published by the instance of the Configuration Service. Each instance of a service publishes multiple interfaces with distinct interface types, and each of these interfaces is represented as a ServiceInstance object. Service instances can be used to register the service with a central configuration service so that other services can use the functionality.

You do not need to configure a database connection to use this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.Configuration.Sdk.ServiceInstance

The Get-ConfigServiceInstance command returns an object containing the following properties.

ServiceGroupUid <Guid>

Specifies the unique identifier for the service group of which the service is a member.

ServiceGroupName <String>

Specifies the name of the service group of which the service is a member.

ServiceInstanceUID <Guid>

Specifies the unique identifier for registered service instances, which are service instances held by and obtained from a



central configuration service. Unregistered service instances do not have unique identifiers.

ServiceType <String>

Specifies the service instance type. For this service, the service instance type is always Config.

Address

Specifies the address of the service instance. The address can be used to access the service and, when registered in the central configuration service, can be used by other services to access the service.

Binding

Specifies the binding type that must be used to communicate with the service instance. In this release of XenDesktop, the binding type is always 'wcf\_HTTP\_kerb'. This indicates that the service provides a Windows Communication Foundation endpoint that uses HTTP binding with integrated authentication.

Version

Specifies the version of the service instance. The version number is used to ensure that the correct versions of the services are used for communications.

ServiceAccount <String>

Specifies the Active Directory account name for the machine on which the service instance is running. The account name is used to provide information about the permissions required for interservice communications.

ServiceAccountSid <String>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

InterfaceType <String>

Specifies the interface type. Each service can provide multiple service instances, each for a different purpose, and the interface defines the purpose. Available interfaces are:

SDK - for PowerShell operations

InterService - for operations between different services

Peer - for communications between services of the same type

Metadata <Citrix.Configuration.Sdk.Metadata[]>

The collection of metadata associated with registered service instances, which are service instances held by and obtained from a central configuration service. Metadata is not stored for unregistered service instances.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-ConfigServiceInstance
```

```
Address      : http://MyServer.com:80/Citrix/ConfigurationService
Binding      : wcf_HTTP_kerb
InterfaceType : SDK
Metadata     :
MetadataMap  :
ServiceAccount : ENG\MyAccount$
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164
ServiceGroupName : MyServiceGroup
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000
ServiceType   : Config
Version      : 1
```

```
Address      : http://MyServer.com:80/Citrix/ConfigurationService/IServiceApi
Binding      : wcf_HTTP_kerb
InterfaceType : InterService
Metadata     :
MetadataMap  :
```

ServiceAccount : ENG\MyAccount  
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164  
ServiceGroupName : MyServiceGroup  
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d  
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000  
ServiceType : Config  
Version : 1

Get all instances of the Configuration Service running on the specified machine. For remote services, use the AdminAddress parameter to define the service for which the interfaces are required. If the AdminAddress parameter has not been specified for the runspace, service instances running on the local machine are returned.

# Get-ConfigServiceStatus

Jan 04, 2017

Gets the current status of the Configuration Service on the controller.

## Syntax

```
Get-ConfigServiceStatus [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables the status of the Configuration Service on the controller to be monitored. If the service has multiple data stores it will return the overall state as an aggregate of all the data store states. For example, if the site data store status is OK and the secondary data store status is DBUnconfigured then it will return DBUnconfigured.

## Related topics

[Set-ConfigDBConnection](#)

[Test-ConfigDBConnection](#)

[Get-ConfigDBConnection](#)

[Get-ConfigDBSchema](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Get-ConfigServiceStatus command returns an object containing the status of the Configuration Service together with extra diagnostics information.

DBUnconfigured

The Configuration Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the Configuration Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

#### InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Configuration Service schema has not been added to the database.

#### DBNotFound

The specified database could not be located with the configured connection string.

#### DBNewerVersionThanService

The version of the Configuration Service currently in use is incompatible with the version of the Configuration Service schema on the database. Upgrade the Configuration Service to a more recent version.

#### DBOlderVersionThanService

The version of the Configuration Service schema on the database is incompatible with the version of the Configuration Service currently in use. Upgrade the database schema to a more recent version.

#### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

#### OK

The Configuration Service is running and is connected to a database containing a valid schema.

#### Failed

The Configuration Service has failed.

#### Unknown

(0) The service status cannot be determined.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigServiceStatus
```

DBUnconfigured

Get the current status of the Configuration Service.

# Get-ConfigSite

Jan 04, 2017

Gets the site.

## Syntax

```
Get-ConfigSite [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Get-ConfigSite cmdlet gets the site.

A XenDesktop installation has only a single site instance.

## Related topics

[Set-ConfigSite](#)

[Import-ConfigFeatureTable](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Get-ConfigSite returns the site instance.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-ConfigSite
```

Gets the site.

# Import-ConfigFeatureTable

Jan 04, 2017

Sets the feature table of the site.

## Syntax

```
Import-ConfigFeatureTable [-Path] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Import-ConfigFeatureTable -Content <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

## Related topics

[Export-ConfigFeatureTable](#)

[Get-ConfigSite](#)

[Set-ConfigSite](#)

[Get-ConfigProduct](#)

[Get-ConfigProductEdition](#)

[Get-ConfigProductFeature](#)

[Get-ConfigProductVersion](#)

[Get-ConfigLicensingModel](#)

## Parameters

**-Path**<String>

The path to the file containing the feature table

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Content**<String>

Set the site's feature table.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |



|                        |                |
|------------------------|----------------|
| Accept Pipeline Input? | true (ByValue) |
|------------------------|----------------|

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Examples

----- **EXAMPLE 1** -----

C:\PS> Import-ConfigFeatureTable \$xml

Specifies the use of a Platinum edition license. A suitable license must be available on the site's license server.

# Register-ConfigServiceInstance

Jan 04, 2017

Allows the registration of a service instance.

## Syntax

```
Register-ConfigServiceInstance -ServiceGroupUid <Guid> -ServiceGroupName <String> -ServiceType <String> -Address <String> -Binding <String> -Version <Int32> -ServiceAccountSid <String> -InterfaceType <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Register-ConfigServiceInstance -ServiceGroupUid <Guid> -ServiceGroupName <String> -ServiceType <String> -Address <String> -Binding <String> -Version <Int32> -ServiceAccount <String> -InterfaceType <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Register-ConfigServiceInstance -ServiceInstance <ServiceInstance[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this cmdlet to register service instance items in the Configuration Service. Service instances can be registered either by retrieving the data directly from other services or by manually entering the details into this command.

If the service group specified by the service instance already exists, the service is added to the service group, otherwise a new service group is created to hold the service instance.

## Related topics

[Unregister-ConfigRegisteredServiceInstance](#)

[Add-ConfigRegisteredServiceInstanceMetadata](#)

[Set-ConfigRegisteredServiceInstanceMetadata](#)

[Remove-ConfigRegisteredServiceInstanceMetadata](#)

[Add-ConfigServiceGroupMetadata](#)

[Set-ConfigServiceGroupMetadata](#)

[Remove-ConfigServiceGroupMetadata](#)

## Parameters

**-ServiceGroupUid**<Guid>

The Service Group Unique Identifier

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

|  |  |
|--|--|
|  |  |
|--|--|

**-ServiceGroupName<String>**

The name of the service group

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ServiceType<String>**

The type of the service group

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Address<String>**

The address that is used to access the service instance.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Binding<String>**

The binding type that must be used to access the service instance.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Version<Int32>**

The version of the service instance.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ServiceAccountSid**<String>

The AD computer account Sid for the computer on which the service instance is hosted.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-InterfaceType**<String>

The type of interface that the service provides (i.e. SDK, InterService, Peer).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ServiceAccount**<String>

The AD computer account name (domain qualified) for the computer on which the service instance is hosted.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ServiceInstance**<ServiceInstance[]>

The service instances to register.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                |
|------------------------|----------------|
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the logging id of the high-level operation that this cmdlet is part of.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Input Type**

Citrix.Configuration.Sdk.ServiceInstance An object with the following parameters can be used to register a service instance. Address, ServiceGroupUid, ServiceGroupName, ServiceType, Binding, Version, InterfaceType.

**Return Values**

Citrix.Configuration.Sdk.ServiceInstance

This represents a service instance and has the following parameters;

ServiceGroupUid <Guid>

The unique identifier for the service group to which the service instance belongs.

ServiceGroupName <string>

The name of the service group to which the service instance belongs.

ServiceInstanceUid <Guid>

The unique identifier for the service instance.

ServiceType <string>

The type of the service group.

Address <string>

The contact address for the service instance.

Binding <string>

The binding to use for connections to the service instance.

Version <int>

The version of the service instance.

ServiceAccount <string>

The AD computer account for the computer that is providing the service instance.

ServiceAccountSid <string>

The AD computer account SID for the computer that is providing the service instance.

InterfaceType <string>

The interface type for the service instance.

Metadata <Citrix.Configuration.Sdk.Metadata[]>

The metadata for the service instance.

## Notes

In the case of failure, the following errors can result.

Error Codes ----- ActiveDirectoryAccountResolutionFailed The account name provided could not be found in Active Directory.

ServiceGroupWithSameUidExistsForDifferentServiceGroupNameOrSameUidExistsForDifferentServiceGroupNameOrServiceType The service group name or service type do not match the service group found with the specified uid. TypeAlreadyExists A different service group with the same type is registered already in the Configuration Service. DatabaseError An error occurred in the service while attempting a database operation. DatabaseNotConfigured The operation could not be completed because the database for the service is not configured. DataStoreException An error occurred in the service while attempting a database operation - communication with the database failed for various reasons. CommunicationError An error occurred while communicating with the service. ExceptionThrown An unexpected error occurred. For more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS>Get-ConfigServiceInstance | Register-ConfigServiceInstance
```

Gets the service instances for the Configuration Service and registers them.

# Remove-ConfigRegisteredServiceInstanceMetadata

Jan 04, 2017

Removes metadata from the given ServiceInstance.

## Syntax

```
Remove-ConfigRegisteredServiceInstanceMetadata [-ServiceInstanceUid] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ConfigRegisteredServiceInstanceMetadata [-ServiceInstanceUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ConfigRegisteredServiceInstanceMetadata [-InputObject] <ServiceInstance[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ConfigRegisteredServiceInstanceMetadata [-InputObject] <ServiceInstance[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given ServiceInstance.

## Related topics

[Add-ConfigRegisteredServiceInstanceMetadata](#)

[Set-ConfigRegisteredServiceInstanceMetadata](#)

## Parameters

**-ServiceInstanceUid**<Guid>

Id of the ServiceInstance

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<ServiceInstance[]>

Objects to which the metadata is to be added.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                |
|------------------------|----------------|
| Accept Pipeline Input? | true (ByValue) |
|------------------------|----------------|

**-Name**<String>

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-ConfigServiceInstance | Remove-ConfigRegisteredServiceInstanceMetadata
```

Remove all metadata from all ServiceInstance objects.

# Remove-ConfigServiceGroup

Jan 04, 2017

Removes service groups.

## Syntax

```
Remove-ConfigServiceGroup [-ServiceGroupUid] <Guid> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Use this cmdlet to remove a service group and all the service instances that it contains.

## Related topics

[Register-ConfigServiceInstance](#)

[Add-ConfigServiceGroupMetadata](#)

[Set-ConfigServiceGroupMetadata](#)

[Remove-ConfigServiceGroupMetadata](#)

## Parameters

**-ServiceGroupUid**<Guid>

The unique identifier for the service group.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the logging id of the high-level operation this cmdlet invocation is part of.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | LocalHost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Notes

In the case of failure, the following errors can result.

**Error Codes** ----- ServiceGroupObjectNotFound The service group specified could not be located. DatabaseError An error occurred in the service while attempting a database operation. DatabaseNotConfigured The operation could not be completed because the database for the service is not configured. DataStoreException An error occurred in the service while attempting a database operation - communication with the database failed for various reasons. CommunicationError An error occurred while communicating with the service. ExceptionThrown An unexpected error occurred. For more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS>Remove-ConfigServiceGroup -ServiceGroupUid 31951f6f-7703-4abb-938a-2861d76ecfea
```

Removes the service group (and all contained service instances) for the service group with a unique identifier of '31951f6f-7703-4abb-938a-2861d76ecfea'.

### ----- EXAMPLE 2 -----

```
C:\PS> Get-configServiceGroup | remove-ConfigServiceGroup
```

Removes all the service groups (and all contained service instances) for the XenDesktop deployment.

# Remove-ConfigServiceGroupMetadata

Jan 04, 2017

Removes metadata from the given ServiceGroup.

## Syntax

```
Remove-ConfigServiceGroupMetadata [-ServiceGroupUid] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ConfigServiceGroupMetadata [-ServiceGroupUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ConfigServiceGroupMetadata [-InputObject] <ServiceGroup[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ConfigServiceGroupMetadata [-InputObject] <ServiceGroup[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given ServiceGroup.

## Related topics

[Add-ConfigServiceGroupMetadata](#)

[Set-ConfigServiceGroupMetadata](#)

## Parameters

**-ServiceGroupUid**<Guid>

Id of the ServiceGroup

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<ServiceGroup[]>

Objects to which the metadata is to be added.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                |
|------------------------|----------------|
| Accept Pipeline Input? | true (ByValue) |
|------------------------|----------------|

**-Name**<String>

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-ConfigServiceGroup | % { Remove-ConfigServiceGroupMetadata -Map $_.MetadataMap }
```

Remove all metadata from all ServiceGroup objects.



# Remove-ConfigServiceMetadata

Jan 04, 2017

Removes metadata from the given Service.

## Syntax

```
Remove-ConfigServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ConfigServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ConfigServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ConfigServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Service.

## Related topics

[Set-ConfigServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Service[]>

Objects to which metadata is to be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Map<PSObject>**

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name<String>**

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId<Guid>**

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigService | % { Remove-ConfigServiceMetadata -Map $_.MetadataMap }  
Remove all metadata from all Service objects.
```

# Remove-ConfigSiteMetadata

Jan 04, 2017

Removes metadata from the given Site.

## Syntax

```
Remove-ConfigSiteMetadata -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ConfigSiteMetadata -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Site.

## Related topics

[Set-ConfigSiteMetadata](#)

## Parameters

**-Name**<String>

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with `@{"name1" = "val1"; "name2" = "val2"}`) or a string dictionary (created with `new-object "System.Collections.Generic.Dictionary[String,String]"`). The properties whose names match keys in the map will be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-

LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigSite | % { Remove-ConfigSiteMetadata -Map $_.MetadataMap }  
Remove all metadata from all Site objects.
```

# Reset-ConfigServiceGroupMembership

Jan 04, 2017

Reloads the access permissions and configuration service locations for the Configuration Service.

## Syntax

```
Reset-ConfigServiceGroupMembership [-ConfigServiceInstance] <ServiceInstance[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables you to reload Configuration Service access permissions and configuration service locations. The Reset-ConfigServiceGroupMembership command must be run on at least one instance of the service type (Config) after installation and registration with the configuration service. Without this operation, the Configuration services will be unable to communicate with other services in the XenDesktop deployment. When the command is run, the services are updated when additional services are added to the deployment, provided that the configuration service is not stopped. The Reset-ConfigServiceGroupMembership command can be run again to refresh this information if automatic updates do not occur when new services are added to the deployment. If more than one configuration service instance is passed to the command, the first instance that meets the expected service type requirements is used.

## Related topics

### Parameters

**-ConfigServiceInstance**<ServiceInstance[]>

Specifies the configuration service instance object that represents the service instance for the type 'InterService' that references a configuration service for the deployment.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | localhost. Once a value is provided by any cmdlet, this value becomes the default. |



|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Input Type

Citrix.Configuration.Sdk.ServiceInstance[] Service instances containing a ServiceInstance object that refers to the central configuration service interservice interface can be piped to the Reset-ConfigServiceGroupMembership command.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoSuitableServiceInstance

None of the supplied service instance objects were suitable for resetting service group membership.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-ConfigServiceGroupMembership
```

Reset the service group membership for a service in a deployment where the configuration service is configured and running on the same machine as the service.

### ----- EXAMPLE 2 -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config -AdminAddress OtherServer.example.com | Reset-ConfigServiceGroupmembership
```

Reset the service group membership for a service in a deployment where the configuration service that is configured and running on a machine named 'OtherServer.example.com'.



# Set-ConfigDBConnection

Jan 04, 2017

Configures a database connection for the Configuration Service.

## Syntax

```
Set-ConfigDBConnection [-DBConnection] <String> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Configures a connection to a database in which the Configuration Service can store its state. The service will attempt to connect and start using the database immediately after the connection is configured. The database connection string is updated to the specified value regardless of whether it is valid or not. Specifying an invalid connection string prevents a service from functioning until the error is corrected.

After a connection is configured, you cannot alter it without first clearing it (by setting the connection to \$null).

You do not need to configure a database connection to use this command.

## Related topics

[Get-ConfigServiceStatus](#)

[Get-ConfigDBConnection](#)

[Test-ConfigDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be used by the Configuration Service. Passing in \$null will clear any existing database connection configured.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Force**<SwitchParameter>

If present, allows the local administrator to set the connection string to null when there are problems contacting the database or other services.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Set-ConfigDBConnection command returns an object containing the status of the Configuration Service together with extra diagnostics information.

DBUnconfigured

The Configuration Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the Configuration Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Configuration Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the Configuration Service currently in use is incompatible with the version of the Configuration Service schema on the database. Upgrade the Configuration Service to a more recent version.

DBOlderVersionThanService

The version of the Configuration Service schema on the database is incompatible with the version of the Configuration Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The Configuration Service is running and is connected to a database containing a valid schema.

## Failed

The Configuration Service has failed.

## Unknown

The status of the Configuration Service cannot be determined.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

### InvalidDBConnectionString

The database connection string has an invalid format.

### DatabaseConnectionDetailsAlreadyConfigured

There was already a database connection configured. After a configuration is set, it can only be set to \$null.

### PermissionDenied

You do not have permission to execute this command.

### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

### CommunicationError

There was a problem communicating with the remote service.

### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-ConfigDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Configures a database connection string for the Configuration Service.

----- **EXAMPLE 2** -----

```
c:\PS>Set-ConfigDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Configures an invalid database connection string for the Configuration Service.

# Set-ConfigRegisteredServiceInstance

Jan 04, 2017

Updates a service instance.

## Syntax

```
Set-ConfigRegisteredServiceInstance -ServiceInstanceUid <Guid> -Address <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

## Detailed Description

Use this cmdlet to change the address property of an existing service instance that is registered in the Configuration Service.

## Related topics

[Get-ConfigRegisteredServiceInstance](#)

[Register-ConfigServiceInstance](#)

[Unregister-ConfigRegisteredServiceInstance](#)

## Parameters

**-ServiceInstanceUid**<Guid>

The unique identifier for the service instance to be updated.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Address**<String>

The new address for the service instance.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

Defines whether or not the command returns a result showing the new state of the updated service instance.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | true  |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the logging id of the high-level operation this cmdlet invocation is part of.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | LocalHost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

Citrix.Configuration.Sdk.ServiceInstance

This represents a service instance and has the following parameters:

**ServiceGroupUid** <Guid>

The unique identifier for the service group to which the service instance belongs.

**ServiceGroupName** <string>

The name of the service group to which the service instance belongs.

**ServiceInstanceUid** <Guid>

The unique identifier for the service instance.

**ServiceType** <string>

The type of the service group.

**Address** <string>

The contact address for the service instance.

**Binding** <string>

The binding to use for connections to the service instance.

**Version** <int>

The version of the service instance.

**ServiceAccount** <string>

The AD computer account for the computer that is providing the service instance.

**ServiceAccountSid** <string>

The AD computer account SID for the computer that is providing the service instance.

**InterfaceType** <string>

The interface type for the service instance.

**Metadata** <Citrix.Configuration.Sdk.Metadata[]>

The metadata for the service instance.

**Notes**

In the case of failure, the following errors can result.

Error Codes ----- ObjectToUpdateDoesNotExist The service instance specified could not be located. DatabaseError An error occurred in the service while attempting a database operation. DatabaseNotConfigured The operation could not be completed because the database for the service is not configured. DataStoreException An error occurred in the service while attempting a database operation - communication with the database failed for various reasons. CommunicationError An error occurred while communicating with the service. ExceptionThrown An unexpected error occurred. For more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

#### Examples

##### ----- **EXAMPLE 1** -----

```
C:\PS>Set-ConfigResgisteredService -ServiceInstanceUid "9805f39d-99eb-44f0-8f63-9d8e3f1228e0" -Address "http://myServer.com/Citrix/sdkHostingUnitService"
```

Update the service instance with the unique identifier of '9805f39d-99eb-44f0-8f63-9d8e3f1228e0' to use the new address.



# Set-ConfigRegisteredServiceInstanceMetadata

Jan 04, 2017

Adds or updates metadata on the given ServiceInstance.

## Syntax

```
Set-ConfigRegisteredServiceInstanceMetadata [-ServiceInstanceUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ConfigRegisteredServiceInstanceMetadata [-ServiceInstanceUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ConfigRegisteredServiceInstanceMetadata [-InputObject] <ServiceInstance[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ConfigRegisteredServiceInstanceMetadata [-InputObject] <ServiceInstance[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given ServiceInstance objects.

## Related topics

[Add-ConfigRegisteredServiceInstanceMetadata](#)

[Remove-ConfigRegisteredServiceInstanceMetadata](#)

## Parameters

**-ServiceInstanceUid**<Guid>

Id of the ServiceInstance

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<ServiceInstance[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                |
|------------------------|----------------|
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the ServiceInstance specified. The property cannot contain any of the following characters \/:#.\*?=<>|[]"

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Direct or typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

System.Collections.Generic.Dictionary[String,String]

Set-ConfigRegisteredServiceInstanceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

### InvalidParameterCombination

The cmdlet parameters are inconsistent.

### UnknownObject

One of the specified objects was not found.

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

### PermissionDenied

You do not have permission to execute this command.

### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

### CommunicationError

There was a problem communicating with the remote service.

### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-ConfigRegisteredServiceInstanceMetadata -ServiceInstanceUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Key      | Value |
|----------|-------|
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the ServiceInstance with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Set-ConfigServiceGroupMetadata

Jan 04, 2017

Adds or updates metadata on the given ServiceGroup.

## Syntax

```
Set-ConfigServiceGroupMetadata [-ServiceGroupUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ConfigServiceGroupMetadata [-ServiceGroupUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ConfigServiceGroupMetadata [-InputObject] <ServiceGroup[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ConfigServiceGroupMetadata [-InputObject] <ServiceGroup[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given ServiceGroup objects.

## Related topics

[Add-ConfigServiceGroupMetadata](#)

[Remove-ConfigServiceGroupMetadata](#)

## Parameters

**-ServiceGroupUid**<Guid>

Id of the ServiceGroup

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Input Object**<ServiceGroup[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the ServiceGroup specified. The property cannot contain any of the following characters \/:;#.\*?=<>|[]()"

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.Collections.Generic.Dictionary[String,String]

Set-ConfigServiceGroupMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-ConfigServiceGroupMetadata -ServiceGroupUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Key      | Value |
|----------|-------|
| ---      | ----  |
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the ServiceGroup with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Set-ConfigServiceMetadata

Jan 04, 2017

Adds or updates metadata on the given Service.

## Syntax

```
Set-ConfigServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ConfigServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ConfigServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ConfigServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Allows you to store additional custom data against given Service objects.

## Related topics

[Remove-ConfigServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Service[]>

Objects to which metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |



**-Map<PObject>**

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with `@{"name1" = "val1"; "name2" = "val2"}`) or a string dictionary (created with `new-object "System.Collections.Generic.Dictionary[String,String]"`).

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name<String>**

Specifies the property name of the metadata to be added. The property must be unique for the Service specified. The property cannot contain any of the following characters `\;#.*?=<>|[]()`

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value<String>**

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId<Guid>**

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the `Start-LogHighLevelOperation` and `Stop-LogHighLevelOperation` cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

System.Collections.Generic.Dictionary[String,String]

Set-ConfigServiceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various

reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-ConfigServiceMetadata -ServiceHostId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Key      | Value |
|----------|-------|
| ---      | ----  |
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the Service with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Set-ConfigSite

Jan 04, 2017

Changes the overall settings of the site.

## Syntax

```
Set-ConfigSite [-SiteName <String>] [-ProductCode <String>] [-ProductEdition <String>] [-ProductVersion <String>] [-LicensingModel <String>] [-LicenseServerName <String>] [-LicenseServerPort <Int32>] [-LicenseServerUri <Uri>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-ConfigSite cmdlet modifies properties of the site.

The site is a top-level, logical representation of the XenDesktop site, from the perspective of the configuration services running within the site.

A XenDesktop installation has only a single site instance.

Modifications to the product code, product edition, product version and licensing model properties are successful only if their values are consistent with the feature table. Use the Get-ConfigProduct, Get-ConfigProductEdition, Get-ConfigProductVersion and Get-ConfigLicensingModel cmdlets to determine consistent values.

To configure the site, first import the feature table using the Import-ConfigFeatureTable cmdlet.

## Related topics

[Export-ConfigFeatureTable](#)

[Get-ConfigSite](#)

[Get-ConfigProduct](#)

[Get-ConfigProductEdition](#)

[Get-ConfigProductFeature](#)

[Get-ConfigProductVersion](#)

[Get-ConfigLicensingModel](#)

## Parameters

**-SiteName**<String>

Changes the name of the site.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-ProductCode**<String>

Changes the product code.

The Get-ConfigProduct cmdlet returns a list of supported values.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ProductEdition**<String>

Changes the license edition. A license matching the specified edition must be available within the site's license server.

The Get-ConfigProductEdition returns a list of supported values.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ProductVersion**<String>

Changes the product version.

The Get-ConfigProductVersion returns a list of supported values.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LicensingModel**<String>

Changes the license model. A license matching the specified model must be available within the site's license server.

The Get-ConfigLicensingModel returns a list of supported values.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LicenseServerName**<String>

Changes the machine used by the brokering services to obtain licenses for desktop and application session brokering. The specified machine must be running a Citrix license server and have suitable licenses installed.

The license server machine can be specified by its DNS name ('machine.domain') or its numeric IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LicenseServerPort**<Int32>

Changes the port number on the license server machine used by the brokering services to contact the Citrix license server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LicenseServerUri**<Uri>

Changes the Uri of the web server for licensing. The hostname component of this Uri must match the Site's LicenseServerName property.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Return Values

Site

Examples

----- **EXAMPLE 1** -----

C:\PS> Set-ConfigSite -ProductEdition PLT

Specifies the use of a Platinum edition license. A suitable license must be available on the site's license server.

# Set-ConfigSiteMetadata

Jan 04, 2017

Adds or updates metadata on the Site.

## Syntax

```
Set-ConfigSiteMetadata -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ConfigSiteMetadata -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against the Site.

## Related topics

[Remove-ConfigSiteMetadata](#)

## Parameters

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Site specified. The property cannot contain any of the following characters \/:;#.\*?=<>|[]()''

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object



"System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

System.Collections.Generic.Dictionary[String,String]

Set-ConfigSiteMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-ConfigSiteMetadata -Name property -Value value
```

| Key | Value |
|-----|-------|
|-----|-------|

---

-----

property

value

Add metadata with a name of 'property' and a value of 'value' to the Site.

# Test-ConfigDBConnection

Jan 04, 2017

Tests a database connection for the Configuration Service.

## Syntax

```
Test-ConfigDBConnection [-DBConnection] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Tests a connection to the database in which the Configuration Service can store its state. The service will attempt to connect to the database without affecting the current connection to the database.

You do not have to clear the connection to use this command.

## Related topics

[Get-ConfigServiceStatus](#)

[Get-ConfigDBConnection](#)

[Set-ConfigDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be tested by the Configuration Service.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Test-ConfigDBConnection command returns an object containing the status of the Configuration Service if the connection string of the specified data store were to be set to the string being tested, together with extra diagnostics information for the specified connection string.

DBRejectedConnection

The database rejected the logon attempt from the Configuration Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Configuration Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the Configuration Service currently in use is incompatible with the version of the Configuration Service schema on the database. Upgrade the Configuration Service to a more recent version.

DBOlderVersionThanService

The version of the Configuration Service schema on the database is incompatible with the version of the Configuration Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The Set-ConfigDBConnection command would succeed if it were executed with the supplied connection string.

Failed

The Configuration Service has failed.

Unknown

The status of the Configuration Service cannot be determined.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

InvalidDBConnectionString

The database connection string has an invalid format.

PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
c:\PS>Test-ConfigDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Tests a database connection string for the Configuration Service.

##### ----- EXAMPLE 2 -----

```
c:\PS>Test-ConfigDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Tests an invalid database connection string for the Configuration Service.

# Test-ConfigServiceInstanceAvailability

Jan 04, 2017

Tests whether the supplied service instances are responding to requests.

## Syntax

```
Test-ConfigServiceInstanceAvailability [-ServiceInstance] <ServiceInstance[]> [-MaxDelaySeconds <Int32>] [-ForceWaitForOneOfEachType] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

## Related topics

[Register-ConfigServiceInstance](#)

[Unregister-ConfigRegisteredServiceInstance](#)

[Get-ConfigRegisteredServiceInstance](#)

## Parameters

**-ServiceInstance**<ServiceInstance[]>

The service instances to test.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-MaxDelaySeconds**<Int32>

The timeout period to wait before concluding that services are unresponsive.

|                        |          |
|------------------------|----------|
| Required?              | false    |
| Default Value          | Infinite |
| Accept Pipeline Input? | false    |

**-ForceWaitForOneOfEachType**<SwitchParameter>

If at least one of each type of service responds, finish immediately.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the host name or IP address of the controller to which the PowerShell snap-in connects.

|                        |                                                                                            |
|------------------------|--------------------------------------------------------------------------------------------|
| Required?              | false                                                                                      |
| Default Value          | 'localhost'. Once a value is specified by any command, this value becomes the new default. |
| Accept Pipeline Input? | false                                                                                      |

**Return Values**

System.Management.Automation.PSObject

This represents a service instance and has the following parameters;

ServiceGroupUid <Guid>

The unique identifier for the service group to which the service instance belongs.

ServiceGroupName <string>

The name of the service group that the service instance is part of.

ServiceInstanceUid <Guid>

The unique identifier for the service instance.

ServiceType <string>

The type of the service group.

Address <string>

The contact address for the service instance.

Binding <string>

The binding to use for connections to the service instance.

Version <int>

The version of the service instance.

ServiceAccount <string>



The AD computer account for the computer that is providing the service instance.

ServiceAccountSid <string>

The AD computer account SID for the computer that is providing the service instance.

InterfaceType <string>

The interface type for the service instance.

Metadata <Citrix.Configuration.Sdk.Metadata[]>

The metadata for the service instance.

Status <Citrix.Configuration.Sdk.Commands.Availability>

An enumeration value indicating whether the service is Responding, NotResponding, Unknown, or BadBindingType.

ResponseTime <System.TineSpan>

The interval elapsed between hailing the service and getting a definite response

## Notes

The Availability Status Codes are  
o Responding: Got a positive response  
o NotResponding: Got a response, but it was negative or the connection was refused  
o Unknown: Did not respond in time / timed-out  
o BadBindingType: Binding parameter in ServiceInstance is not wcf\_HTTP\_kerb

## Examples

### ----- EXAMPLE 1 -----

```
C:\>Get-ConfigRegisteredServiceInstance | Test-ConfigServiceInstanceAvailability -ForceWaitForOneOfEachType
```

Test all the service instances that are registered in the Configuration Service, returning when one of each type is responding.

### ----- EXAMPLE 2 -----

```
C:\>Test-ConfigServiceInstanceAvailability -ServiceInstance $services -MaxDelaySeconds 5
```

Test each of the given services, allowing a 5 second time-out.

# Unregister-ConfigRegisteredServiceInstance

Jan 04, 2017

Removes a service instance from the Configuration Service registry.

## Syntax

```
Unregister-ConfigRegisteredServiceInstance [-ServiceInstanceId] <Guid> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Use this cmdlet to remove a service instance from the Configuration Service registry. This does not remove any service groups (if all service instances for a Service Group are removed, an empty service group remains and must be removed using the Remove-ConfigServiceGroup command).

## Related topics

[Register-ConfigServiceInstance](#)

## Parameters

**-ServiceInstanceId**<Guid>

The unique identifier for the service instance to be removed.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the logging id of the high-level operation this cmdlet invocation is part of.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | LocalHost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Input Type

Citrix.Configuration.Sdk.ServiceInstance An object with a parameter called 'ServiceInstanceUid' can be used to unregister service instances.

## Notes

In the case of failure, the following errors can result.

Error Codes ----- ServiceInstanceObjectNotFound The service instance specified could not be located. DatabaseError An error occurred in the service while attempting a database operation. DatabaseNotConfigured The operation could not be completed because the database for the service is not configured. DataStoreException An error occurred in the service while attempting a database operation - communication with the database failed for various reasons. CommunicationError An error occurred while communicating with the service. ExceptionThrown An unexpected error occurred. For more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

C:\PS>Get-ConfigRegisteredServiceInstance -ServiceType "Config" | Unregister-ConfigRegisteredServiceInstance  
 Unregisters all the service instances that are for a service type of 'Config' from the Configuration Service instance register.

# Citrix.ConfigurationLogging.Admin.V1

Jan 04, 2017  
Overview

| Name                                          | Description                                                                  |
|-----------------------------------------------|------------------------------------------------------------------------------|
| <a href="#">LogConfigurationLoggingSnapin</a> | The Configuration Logging Service PowerShell snap-in provides administrative |
| <a href="#">Log Filtering</a>                 | Describes the common filtering options for XenDesktop cmdlets.               |
| <a href="#">logical operators</a>             | Describes the operators that connect statements in Windows PowerShell.       |

## Cmdlets

| Name                                          | Description                                                                                               |
|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| <a href="#">Export-LogReportCsv</a>           | Exports Configuration Logging data into a CSV file.                                                       |
| <a href="#">Export-LogReportHtml</a>          | Exports Configuration Logging data into a HTML report.                                                    |
| <a href="#">Get-LogDataStore</a>              | Gets details for each of the ConfigurationLogging data stores.                                            |
| <a href="#">Get-LogDBConnection</a>           | Gets the database string for the specified data store used by the ConfigurationLogging Service.           |
| <a href="#">Get-LogDBSchema</a>               | Gets a script that creates the ConfigurationLogging Service database schema for the specified data store. |
| <a href="#">Get-LogDBVersionChangeScript</a>  | Gets a script that updates the ConfigurationLogging Service database schema.                              |
| <a href="#">Get-LogHighLevelOperation</a>     | Gets high level operations                                                                                |
| <a href="#">Get-LogInstalledDBVersion</a>     | Gets a list of all available database schema versions for the ConfigurationLogging Service.               |
| <a href="#">Get-LogLowLevelOperation</a>      | Gets low level operations                                                                                 |
| <a href="#">Get-LogService</a>                | Gets the service record entries for the ConfigurationLogging Service.                                     |
| <a href="#">Get-LogServiceAddedCapability</a> | Gets any added capabilities for the ConfigurationLogging Service on the controller.                       |

| Name                                            | Description                                                                                              |
|-------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <a href="#">Get-LogServiceInstance</a>          | Gets the service instance entries for the ConfigurationLogging Service.                                  |
| <a href="#">Get-LogServiceStatus</a>            | Gets the current status of the ConfigurationLogging Service on the controller.                           |
| <a href="#">Get-LogSite</a>                     | Gets global configuration logging settings.                                                              |
| <a href="#">Get-LogSummary</a>                  | Gets operations logged within time intervals inside a date range.                                        |
| <a href="#">Remove-LogOperation</a>             | Deletes configuration logs                                                                               |
| <a href="#">Remove-LogServiceMetadata</a>       | Removes metadata from the given Service.                                                                 |
| <a href="#">Remove-LogSiteMetadata</a>          | Removes metadata from the given Site.                                                                    |
| <a href="#">Reset-LogDataStore</a>              | Refreshes the database string currently being used by the Log service.                                   |
| <a href="#">Reset-LogServiceGroupMembership</a> | Reloads the access permissions and configuration service locations for the ConfigurationLogging Service. |
| <a href="#">Set-LogDBConnection</a>             | Configures a database connection for the ConfigurationLogging Service.                                   |
| <a href="#">Set-LogServiceMetadata</a>          | Adds or updates metadata on the given Service.                                                           |
| <a href="#">Set-LogSite</a>                     | Sets global configuration logging settings.                                                              |
| <a href="#">Set-LogSiteMetadata</a>             | Adds or updates metadata on the given Site.                                                              |
| <a href="#">Start-LogHighLevelOperation</a>     | Logs the start of a high level operation.                                                                |
| <a href="#">Stop-LogHighLevelOperation</a>      | Logs the completion of a previously started high level operation.                                        |
| <a href="#">Test-LogDBConnection</a>            | Tests a database connection for the ConfigurationLogging Service.                                        |

# about\_LogConfigurationLoggingSnapin

Jan 04, 2017

## TOPIC

about\_LogConfigurationLoggingSnapin

## SHORT DESCRIPTION

The Configuration Logging Service PowerShell snap-in provides administrative functions for the Configuration Logging Service.

## COMMAND PREFIX

All commands in this snap-in have the noun prefixed with 'Log'.

## LONG DESCRIPTION

The Configuration Logging Service PowerShell snap-in enables both local and remote administration of the Configuration Logging Service.

The Configuration Logging Service logs configuration changes or administrator requested state changes made to the site. Configuration Logging can be configured, site wide, to be mandatory or optional. If mandatory logging is selected, then any attempts to change site configuration or state when the logging mechanism is unavailable are denied.

The Configuration Logging Service stores information about the logged changes in a database which can be configured to be separate from the site database.

The snap-in provides storage and configuration of these entities:

### Site

The Configuration Logging Site object holds global settings which control the behaviour of the Configuration Logging Service. The site object can be configured by the Set-LogSite cmdlet. The properties of the site object are returned by the Get-LogSite cmdlet.

## High Level Operations

A high level operation object represents a logged configuration change performed from Desktop Studio, Desktop Director or a PowerShell Script.

The XenDesktop consoles log high level operations when:

- 1) Executing operations which performs configuration changes.
- 2) Executing operations which performs administration related activities which may affect site configuration.

PowerShell scripts which carry out customized configuration changes can also log high level operations via cmdlets `Start-LogHighLevelOperation` and `Stop-LogHighLevelOperation`.

## Low Level Operations

A low level operation object represents a logged configuration change performed by a service. One or more low level operation objects are used to record the actions performed by a services in order to fulfil a high level operation initiated from the consoles, or from PowerShell scripts.

Low level operations in the system are returned by cmdlet `Get-LogLowLevelOperation`.

## Operation Details

A low level operation performed by a service can affect a number of individual objects, or a number of properties on an object. An operation detail log records each individual change to an object. This includes the creation and deletion of the object, as well as changes to individual properties of the object.

One or more operation detail objects are used to record specific changes to each object that is affected by a low level service operation.

Operation details are included in the data returned from the `Get-LogLowLevelOperation` cmdlet.

High Level Operations, Low Level Operations and Operation Details are arranged in a hierarchy. A High Level Operation can have multiple Low Level Operations, and each Low Level Operation can have multiple Operation Details.

Setting up a separate logging database ----- After creating the database on the database server, the logging database can be setup and configured for use by:

- 1) Generating the database schema, and applying it the logging database
- 2) Configuring the configuration logging service to use the new logging database.

The logging database schema can be generated from the `Get-LogDBSchem` cmdlet, as illustrated below:

```
Get-LogDBSchema -DatabaseName "loggingDB"  
  -ServiceGroupName "service group name"  
  -ScriptType Database  
  -LocalDatabase:$LocalDB  
  -DataStore Logging
```

The configuration logging service can be configured to use the logging database with the Set-LogDBConnection cmdlet, as illustrated below:

```
Set-LogDBConnection -DataStore Logging -DBConnection $null  
Set-LogDBConnection -DataStore Logging -DBConnection "new logging db connection string"
```

Configuration Logging site settings -----

On the Site object:

- o The 'State' setting allows configuration logging to be disabled, enabled, or made mandatory.
- o The 'Locale' setting specifies the language in which configuration logging data text will be stored.

See Get-LogSite cmdlet help for further information on these settings.

This locale setting applies to the text description that is associated with each log, e.g. **Create Catalog**. It doesn't apply to other textual information in the log like the names of parameters passed to operations, e.g. **CatalogName**.

This localisation is applied when the data is logged, and not when the logs are viewed later. For example, logs which are created in English will be displayed in English on an end user system which may be configured with a different locale.



# about\_Log\_Filtering

Jan 04, 2017

## TOPIC

XenDesktop - Advanced Dataset Filtering

## SHORT DESCRIPTION

Describes the common filtering options for XenDesktop cmdlets.

## LONG DESCRIPTION

Some cmdlets operate on large quantities of data and, to reduce the overhead of sending all of that data over the network, many of the Get- cmdlets support server-side filtering of the results.

The conventional way of filtering results in PowerShell is to pipeline them into Where-Object, Select-Object, and Sort-Object, for example:

```
Get-<Noun> | Where { $_.Size = 'Small' } | Sort 'Date' | Select -First 10
```

However, for most XenDesktop cmdlets the data is stored remotely and it would be slow and inefficient to retrieve large amounts of data over the network and then discard most of it. Instead, many of the Get- cmdlets provide filtering parameters that allow results to be processed on the server, returning only the required results.

You can filter results by most object properties using parameters derived from the property name. You can also sort results or limit them to a specified number of records:

```
Get-<Noun> -Size 'Small' -SortBy 'Date' -MaxRecordCount 10
```

You can express more complex filter conditions using a syntax and set of operators very similar to those used by PowerShell expressions.

Those cmdlets that support filtering have the following common parameters:

-MaxRecordCount <int>

Specifies the maximum number of results to return.  
For example, to return only the first nine results use:

```
Get-<Noun> -MaxRecordCount 9
```

If not specified, only the first 250 records are returned, and if more are available, a warning is produced:

WARNING: Only first 250 records returned. Use -MaxRecordCount to retrieve more.

You can suppress this warning by using -WarningAction or by specifying a value for -MaxRecordCount.

To retrieve all records, specify a large number for -MaxRecordCount. As the value is an integer, you can use the following:

```
Get-<Noun> -MaxRecordCount [int]::MaxValue
```

-ReturnTotalRecordCount [<SwitchParameter>]

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. For example:

```
Get-<Noun> -MaxRecordCount 9 -ReturnTotalRecordCount
....
```

```
Get-<Noun> : Returned 9 of 10 items
At line:1 char:18
+ Get-<Noun> <<<< -MaxRecordCount 9 -ReturnTotalRecordCount
+ CategoryInfo          : OperationStopped: (:) [Get-<Noun>], PartialDataException
+ FullyQualifiedErrorId : PartialData,Citrix.<SDKName>.SDK.Get<Noun>
```

The count can be accessed using the TotalAvailableResultCount property:

```
$count = $error[0].TotalAvailableResultCount
```

-Skip <int>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

-SortBy <string>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order, respectively. Ascending order is assumed if no prefix is present.

Sorting occurs before `-MaxRecordCount` and `-Skip` parameters are applied. For example, to sort by Name and then by Count (largest first) use:

```
-SortBy 'Name,-Count'
```

By default, sorting by an enumeration property uses the numeric value of the elements. You can specify a different sort order by qualifying the name with an ordered list of elements or their numeric values, or `<null>` to indicate the placement of null values.

Elements not mentioned are placed at the end in their numeric order. For example, to sort by two different enums and then by the object id:

```
-SortBy 'MyState(StateC,<null>,StateA,StateB),Another(0,3,2,1),Id'
```

`-Filter <String>`

This parameter lets you specify advanced filter expressions, and supports combination of conditions with `-and` and `-or`, and grouping with braces. For example:

```
Get-<Noun> -Filter 'Name -like "High*" -or (Priority -eq 1 -and Severity -ge 2)'
```

The syntax is close enough to PowerShell syntax that you can use script blocks in most cases. This can be easier to read as it reduces quoting:

```
Get-<Noun> -Filter { Count -ne $null }
```

The full `-Filter` syntax is provided below.

## EXAMPLES

Filtering by strings performs a case-insensitive wildcard match.

Separate parameters are combined with an implicit `-and` operator.

Normal PowerShell quoting rules apply, so you can use single or double quotes, and omit the quotes altogether for many strings. The order of parameters does not make any difference. The following are equivalent:

```
Get-<Noun> -Company Citrix -Product Xen*
Get-<Noun> -Company "citrix" -Product '[X]EN*'
Get-<Noun> -Product "Xen*" -Company "CITRIX"
Get-<Noun> -Filter { Company -eq 'Citrix' -and Product -like 'Xen*' }
```

See [about\\_Quoting\\_Rules](#) and [about\\_Wildcards](#) for details about PowerShell handling of quotes and wildcards.

To avoid wildcard matching or include quote characters, you can escape the wildcards using the normal PowerShell escape mechanisms (see [about\\_Escape\\_Characters](#)), or switch to a filter expression and the `-eq` operator:

```
Get-<Noun> -Company "Abc[*]"           # Matches Abc*
Get-<Noun> -Company "Abc`*"           # Matches Abc*
Get-<Noun> -Filter { Company -eq "Abc*" } # Matches Abc*
Get-<Noun> -Filter { Company -eq "A`"B`C" } # Matches A"B'C
```

Simple filtering by numbers, booleans, and TimeSpans perform direct equality comparisons, although if the value is nullable you can also search for null values. Here are some examples:

```
Get-<Noun> -Uid 123
Get-<Noun> -Enabled $true
Get-<Noun> -Duration 1:30:40
Get-<Noun> -NullableProperty $null
```

More comparisons are possible using advanced filtering with `-Filter`:

```
Get-<Noun> -Filter 'Capacity -ge 10gb'
Get-<Noun> -Filter 'Age -ge 20 -and Age -lt 40'
Get-<Noun> -Filter 'VolumeLevel -like "[123]"'
Get-<Noun> -Filter 'Enabled -ne $false'
Get-<Noun> -Filter 'NullableProperty -ne $null'
```

You can check boolean values without an explicit comparison operator, and you can also combine them with `-not`:

```
Get-<Noun> -Filter 'Enabled' # Equivalent to 'Enabled -eq $true'
Get-<Noun> -Filter '-not Enabled' # Equivalent to 'Enabled -eq $false'
```

See [about\\_Comparison\\_Operators](#) for an explanation of the operators, but note that only a subset of PowerShell operators are supported (`-eq`, `-ne`, `-gt`, `-ge`, `-lt`, `-le`, `-like`, `-notlike`, `-in`, `-notin`, `-contains`, `-notcontains`).

Enumeration values can either be specified using typed values or the string name of the enumeration value:

```
Get-<Noun> -Shape [Shapes]::Square
Get-<Noun> -Shape Circle
```

With filter expressions, typed values can be specified with simple variables or quoted strings. They also support enumerations with wildcards:

```
$s = [Shapes]::Square
Get-<Noun> -Filter { Shape -eq $s -or Shape -eq "Circle" }
Get-<Noun> -Filter { Shape -like 'C*' }
```

By their nature, floating point values, DateTime values, and TimeSpan values are best suited to relative comparisons rather than just equality. DateTime strings are converted using the locale and time zone of the user device, but you can use ISO8601 format strings (YYYY-MM-DDThh:mm:ss.sTZD) to avoid ambiguity. You can also use standard PowerShell syntax to create these values:

```
Get-<Noun> -Filter { StartTime -ge "2010-08-23T12:30:00.0Z" }
$d = [DateTime]"2010-08-23T12:30:00.0Z"
Get-<Noun> -Filter { StartTime -ge $d }
$d = (Get-Date).AddDays(-1)
Get-<Noun> -Filter { StartTime -ge $d }
```

Relative times are quite common and, when using filter expressions, you can also specify DateTime values using a relative format:

```
Get-<Noun> -Filter { StartTime -ge '-2' } # Two days ago
Get-<Noun> -Filter { StartTime -ge '-1:30' } # Hour and a half ago
Get-<Noun> -Filter { StartTime -ge '-0:0:30' } # 30 seconds ago
```

## ARRAY PROPERTIES

When filtering against list or array properties, simple parameters perform a case-insensitive wildcard match against each of the members. With filter expressions, you can use the -contains and -notcontains operators. Unlike PowerShell, these perform wildcard matching on strings.

Note that for array properties the naming convention is for the returned property to be plural, but the parameter used to search for any match is singular. The following are equivalent (assuming Users is an array property):

```
Get-<Noun> -User Fred*
Get-<Noun> -Filter { User -like "Fred*" }
Get-<Noun> -Filter { Users -contains "Fred*" }
```

You can also use the singular form with -Filter to search using other operators:

```
# Match if any user in the list is called "Frederick"
Get-<Noun> -Filter { User -eq "Frederick" }
# Match if any user in the list has a name alphabetically below 'F'
Get-<Noun> -Filter { User -lt 'F' }
```

## COMPLEX EXPRESSIONS

When matching against multiple values, you can use a sequence of comparisons joined with -or operators, or you can use -in and -notin:

```
Get-<Noun> -Filter { Shape -eq 'Circle' -or Shape -eq 'Square' }
$shapes = 'Circle','Square'
Get-<Noun> -Filter { Shape -in $shapes }
$sides = 1..4
Get-<Noun> -Filter { Sides -notin $sides }
```

Braces can be used to group complex expressions, and override the default left-to-right evaluation of -and and -or. You can also use -not to invert the sense of any sub-expression:

```
Get-<Noun> -Filter { Size -gt 4 -or (Color -eq 'Blue' -and Shape -eq 'Circle') }
Get-<Noun> -Filter { Sides -lt 5 -and -not (Color -eq 'Blue' -and Shape -eq 'Circle') }
```

## PAGING

The simplest way to page through data is to use the -Skip and -MaxRecordCount parameters. So, to read the first three pages of data with 10 records per page, use:

```
Get-<Noun> -Skip 0 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 10 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 20 -MaxRecordCount 10 <other filtering criteria>
```

You must include the same filtering criteria on each call, and ensure that the data is sorted consistently.

The above approach is often acceptable, but as each call performs an independent query, data changes can result in records being skipped or appearing twice. One approach to improve this is to sort by a unique id field and then start the search for the next page at the unique id after the last unique id of the previous page. For example:

```
# Get the first page
Get-<Noun> -MaxRecordCount 10 -SortBy SerialNumber

SerialNumber ...
----- ---
A120004
A120007
... 7 other records ...
A120900

# Get the next page
Get-<Noun> -MaxRecordCount 10 -Filter { FirstName -gt 'A120900' }

SerialNumber ...
```

----- ---  
A120901  
B220000  
...

## FILTER SYNTAX DEFINITION

<Filter> ::= <ScriptBlock> | <ComponentList>

<ScriptBlock> ::= "{" <ComponentList> "}"

<ComponentList> ::= <Component> <AndOrOperator> <ComponentList> |

<Component>

<Component> ::= <NotOperator> <Factor> |

<Factor>

<Factor> ::= "(" <ComponentList> ")" |

<PropertyName> <ComparisonOperator> <Value> |  
<PropertyName>

<AndOrOperator> ::= "-and" | "-or"

<NotOperator> ::= "-not" | "!"

<ComparisonOperator>

::= "-eq" | "-ne" | "-le" | "-ge" | "-lt" | "-gt" |  
"-like" | "-notlike" | "-contains" | "-notcontains" |  
"-in" | "-notin"

<PropertyName> ::= <simple name of property>

<Value> ::= <string literal> | <numeric literal> |

<scalar variable> | <array variable> |  
"\$null" | "\$true" | "\$false"

Numeric literals support decimal and hexadecimal literals, with optional multiplier suffixes (kb, mb, gb, tb, pb).

Dates and times can be specified as string literals. The current culture determines what formats are accepted. To avoid any ambiguity, use strings formatted to the ISO8601 standard. If not specified, the current time zone is used.

Relative date-time string literals are also supported, using a minus sign followed by a TimeSpan. For example, "-1:30" means 1 hour and 30 minutes ago.



# about\_logical\_operators

Jan 04, 2017

## TOPIC

about\_Logical\_Operators

## SHORT DESCRIPTION

Describes the operators that connect statements in Windows PowerShell.

## LONG DESCRIPTION

The Windows PowerShell logical operators connect expressions and statements, allowing you to use a single expression to test for multiple conditions.

For example, the following statement uses the and operator and the or operator to connect three conditional statements. The statement is true only when the value of \$a is greater than the value of \$b, and either \$a or \$b is less than 20.

```
($a -gt $b) -and (($a -lt 20) -or ($b -lt 20))
```

Windows PowerShell supports the following logical operators.

| Operator | Description                                                                                | Example                           |
|----------|--------------------------------------------------------------------------------------------|-----------------------------------|
| -and     | Logical and. TRUE only when both statements are TRUE.                                      | (1 -eq 1) -and (1 -eq 2)<br>False |
| -or      | Logical or. TRUE when either or both statements are TRUE.                                  | (1 -eq 1) -or (1 -eq 2)<br>True   |
| -xor     | Logical exclusive or. TRUE only when one of the statements is TRUE and the other is FALSE. | (1 -eq 1) -xor (2 -eq 2)<br>False |
| -not     | Logical not. Negates the statement that follows it.                                        | -not (1 -eq 1)<br>False           |
| !        | Logical not. Negates the statement that follows it. (Same as -not)                         | !(1 -eq 1)<br>False               |

Note: The previous examples also use the equal to comparison

operator (-eq). For more information, see [about\\_Comparison\\_Operators](#).  
The examples also use the Boolean values of integers. The integer 0 has a value of FALSE. All other integers have a value of TRUE.

The syntax of the logical operators is as follows:

```
<statement> {-AND | -OR | -XOR} <statement>  
{! | -NOT} <statement>
```

Statements that use the logical operators return Boolean (TRUE or FALSE) values.

The Windows PowerShell logical operators evaluate only the statements required to determine the truth value of the statement. If the left operand in a statement that contains the and operator is FALSE, the right operand is not evaluated. If the left operand in a statement that contains the or statement is TRUE, the right operand is not evaluated. As a result, you can use these statements in the same way that you would use the If statement.

#### SEE ALSO

[about\\_Operators](#)

[Compare-Object](#)

[about\\_Comparison\\_operators](#)

[about\\_If](#)

# Export-LogReportCsv

Jan 04, 2017

Exports Configuration Logging data into a CSV file.

## Syntax

```
Export-LogReportCsv -OutputFile <String> [-StartDateRange <DateTime>] [-EndDateRange <DateTime>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet exports the Configuration Logging data into a CSV data file. The hierarchical logging data is flattened into a single CSV 'table'. The content of CSV file is not intended to be human-readable. It is meant to be input data for external reporting or manipulation tools (for example, a spread sheet application).

## Related topics

[Export-LogReportHtml](#)

## Parameters

### **-OutputFile**<String>

Specifies the path to a file where the CSV data will be saved.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-StartDateRange**<DateTime>

Specifies the start time of the earliest operation to include.

|                        |              |
|------------------------|--------------|
| Required?              | false        |
| Default Value          | DateTime.Min |
| Accept Pipeline Input? | false        |

### **-EndDateRange**<DateTime>

Specifies the end time of the latest operation to include.

|                        |                 |
|------------------------|-----------------|
| Required?              | false           |
| Default Value          | DateTime.UtcNow |
| Accept Pipeline Input? | false           |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Export-LogReportCsv -OutputFile "c:\MyReports\LoggingData.csv"
```

Export all logged operations to a csv file.

----- **EXAMPLE 2** -----

```
C:\PS> Export-LogReportCsv -OutputFile "c:\MyReports\LoggingData.csv" -StartDateRange "2012-12-21 09:00"
```

Export to a CSV file logged operations started on or after a specified datetime..

----- **EXAMPLE 3** -----

```
C:\PS> Export-LogReportCsv -OutputFile "c:\MyReports\LoggingData.csv" -StartDateRange "2012-12-21 09:00" -EndDateRange "2012-12-31 18:00"
```

Export to a CSV file logged operations started and completed between a date range.

# Export-LogReportHtml

Jan 04, 2017

Exports Configuration Logging data into a HTML report.

## Syntax

```
Export-LogReportHtml -OutputDirectory <String> [-StartDateRange <DateTime>] [-EndDateRange <DateTime>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet exports the Configuration Logging data into a HTML report. The report consists of two HTML files:

- o Summary.Html - this shows summary information from the high level operation logs.
- o Details.Html - this shows additional logging data from the low level operation and operation detail logs.

Hyperlinks in summary.html allow drill-down into the associated low level logging data contained within details.html.

## Related topics

[Export-LogReportCsv](#)

## Parameters

**-OutputDirectory**<String>

Specifies the path to a directory where the HTML report files will be saved.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartDateRange**<DateTime>

Specifies the start time of the earliest operation to include.

|                        |              |
|------------------------|--------------|
| Required?              | false        |
| Default Value          | DateTime.Min |
| Accept Pipeline Input? | false        |

**-EndDateRange**<DateTime>

Specifies the end time of the latest operation to include.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                 |
|------------------------|-----------------|
| Default Value          | DateTime.UtcNow |
| Accept Pipeline Input? | false           |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Export-LogReportHtml -OutputDirectory "c:\MyReports"
Export all logged operations to HTML.
```

----- **EXAMPLE 2** -----

```
C:\PS> Export-LogReportHtml -OutputDirectory "c:\MyReports" -StartDateRange "2012-12-21 09:00"
Export to a HTML logged operations started on or after a specified datetime..
```

----- **EXAMPLE 3** -----

```
C:\PS> Export-LogReportHtml -OutputDirectory "c:\MyReports" -StartDateRange "2012-12-21 09:00" -EndDateRange "2012-12-31 18:00"
Export to HTML logged operations started and completed between a date range.
```

# Get-LogDataStore

Jan 04, 2017

Gets details for each of the ConfigurationLogging data stores.

## Syntax

```
Get-LogDataStore [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns an object for each of the ConfigurationLogging data stores describing the connection string, data store name, db type, provider, schema name, and DB status.

A database connection must be configured in order for this command to be used if the service has a secondary data store. This is not required for the site data store.

## Related topics

[Reset-LogDataStore](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.ConfigurationLogging.Sdk.DataStoreConfiguration

An object describing the connection string, data store name, database type, provider, schema name and database status.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

### DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-LogDataStore
```

```
ConnectionString : Server=.\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True
DataStore      : Site
DatabaseType   : SqlServer
Provider       : MSSQL
SchemaName     : LogSiteSchema
Status        : OK
```

```
ConnectionString :
DataStore      : Secondary
DatabaseType   : SqlServer
Provider       : MSSQL
SchemaName     : LogSecondarySchema
Status        : DBUnconfigured
```

Get the database connection string for the ConfigurationLogging Service.



# Get-LogDBConnection

Jan 04, 2017

Gets the database string for the specified data store used by the ConfigurationLogging Service.

## Syntax

```
Get-LogDBConnection [[-DataStore] <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the database connection string for the specified data store.

If the returned string is blank, no valid connection string has been specified. In this case the service is running, but is idle and awaiting specification of a valid connection string.

## Related topics

[Get-LogServiceStatus](#)

[Get-LogDataStore](#)

[Set-LogDBConnection](#)

[Test-LogDBConnection](#)

## Parameters

**-DataStore**<String>

Specifies the logical name of the data store for the ConfigurationLogging Service. Can be either be 'Site' or the logical name of the secondary data store.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | Site  |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

system.string

The database connection string configured for the ConfigurationLogging Service.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoDBConnections

The database connection string for the ConfigurationLogging Service has not been specified.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-LogDBConnection
```

Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True  
Get the database connection string for the ConfigurationLogging Service.

# Get-LogDBSchema

Jan 04, 2017

Gets a script that creates the ConfigurationLogging Service database schema for the specified data store.

## Syntax

```
Get-LogDBSchema [-DatabaseName <String>] [-ServiceGroupName <String>] [-ScriptType <ScriptTypes>] [-LocalDatabase] [-Sid <String>] [-DataStore <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets SQL scripts that can be used to create a new ConfigurationLogging Service database schema, add a new ConfigurationLogging Service to an existing site, remove a ConfigurationLogging Service from a site, or create a database server logon for a ConfigurationLogging Service. If no Sid parameter is provided, the scripts obtained relate to the currently selected ConfigurationLogging Service instance, otherwise the scripts relate to ConfigurationLogging Service instance running on the machine identified by the Sid provided. When obtaining the Evict script, a Sid parameter must be supplied. The current service instance is that on the local machine, or that explicitly specified by the last usage of the -AdminAddress parameter to a ConfigurationLogging SDK cmdlet. The service instance used to obtain the scripts does not need to be a member of a site or to have had its database connection configured. The database scripts support only Microsoft SQL Server, or SQL Server Express, and require Windows integrated authentication to be used. They can be run using SQL Server's SQLCMD utility, or by copying the script into an SQL Server Management Studio (SSMS) query window and executing the query. If using SSMS, the query must be executed in 'SMDCMD mode'. The ScriptType parameter determines which script is obtained. If ScriptType is not specified, or is FullDatabase, the script contains:

- o Creation of service schema
- o Creation of database server logon
- o Creation of database user
- o Addition of database user to ConfigurationLogging Service roles

If ScriptType is Instance, the returned script contains:

- o Creation of database server logon
- o Creation of database user
- o Addition of database user to ConfigurationLogging Service roles

If ScriptType is Evict, the returned script contains:

- o Removal of ConfigurationLogging Service instance from database
- o Removal of database user

If ScriptType is Login, the returned script contains:

- o Creation of database server logon only

If the service uses two data stores they can exist in the same database. You do not need to configure a database before using this command.

## Related topics

## Get-LogDataStore

## Set-LogDBConnection

### Parameters

#### **-DatabaseName**<String>

Specifies the name of the database for which the schema will be generated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-ServiceGroupName**<String>

Specifies the name of the service group to be used when creating the database schema. The service group is a collection of all the ConfigurationLogging services that share the same database instance and are considered equivalent; that is, all the services within a service group can be used interchangeably.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-ScriptType**<ScriptTypes>

Specifies the type of database script returned. Available script types are:

##### Database

Returns a full database script that can be used to create a database schema for the ConfigurationLogging Service in a database instance that does not already contain a schema for this service. The DatabaseName and ServiceGroupName parameters must be specified to create a script of this type.

##### Instance

Returns a permissions script that can be used to add further ConfigurationLogging services to an existing database instance that already contains the full ConfigurationLogging service schema, associating the services to the Service Group. The Sid parameter can optionally be specified to create a script of this type.

##### Login

Returns a database logon script that can be used to add the required logon accounts to an existing database instance that contains the ConfigurationLogging Service schema. This is used primarily when creating a mirrored database environment. The DatabaseName parameter must be specified to create a script of this type.

## Evict

Returns a script that can be used to remove the specified ConfigurationLogging Service from the database entirely. The DatabaseName and Sid parameters must be specified to create a script of this type.

|                        |          |
|------------------------|----------|
| Required?              | false    |
| Default Value          | Database |
| Accept Pipeline Input? | false    |

## -LocalDatabase<SwitchParameter>

Specifies whether the database script is to be used in a database instance run on the same controller as other services in the service group. Including this parameter ensures the script creates only the required permissions for local services to access the database schema for ConfigurationLogging services.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## -Sid<String>

Specifies the SID of the controller on which the ConfigurationLogging Service instance to remove from the database is running.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

## -DataStore<String>

Specifies the logical name of the data store for the ConfigurationLogging Service. Can be either be 'Site' or the logical name of the secondary data store.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | Site  |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

System.string

A string containing the required SQL script for application to a database.

### Notes

The scripts returned support Microsoft SQL Server Express Edition, Microsoft SQL Server Standard Edition, and Microsoft SQL Server Enterprise Edition databases only, and are generated on the assumption that integrated authentication will be used.

If the ScriptType parameter is not included or set to 'FullDatabase', the full database script is returned, which will:

Create the database schema.

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist).

If the ScriptType parameter is set to 'Instance', the script will:

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist) and associate it with a user.

If the ScriptType parameter is set to 'Login', the script will:

Create the logon (providing the schema does not already exist) and associate it with a pre-existing user of the same name.

If the LocalDatabase parameter is included, the NetworkService account will be added to the list of accounts permitted to access the database. This is required only if the database is run on a controller.

If the command fails, the following errors can be returned.

### Error Codes

-----

GetSchemasFailed

The database schema could not be found.

ActiveDirectoryAccountResolutionFailed

The specified Active Directory account or Group could not be found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-LogDBSchema -DatabaseName MyDB -ServiceGroupName MyServiceGroup > c:\LogSchema.sql
Get the full database schema for site data store of the ConfigurationLogging Service and copy it to a file called 'c:\LogSchema.sql'.
```

This script can then be used to create the schema in a pre-existing database named 'MyDB' that does not already contain a ConfigurationLogging Service site schema.

----- **EXAMPLE 2** -----

```
c:\PS>Get-LogDBSchema -DatabaseName MyDB -scriptType Login > c:\ConfigurationLoggingLogins.sql
Get the logon scripts for the ConfigurationLogging Service.
```

----- **EXAMPLE 3** -----

```
c:\PS>Get-LogDBSchema -DatabaseName MyDB -ServiceGroupName MyServiceGroup -DataStore Secondary > c:\LogSchema.sql
Get the full database schema for the secondary data store of the ConfigurationLogging Service and copy it to a file called 'c:\LogSecondarySchema.sql'.
```

This script can then be used to create the schema in a pre-existing database named 'MyDB' that does not already contain a ConfigurationLogging Service secondary schema.



# Get-LogDBVersionChangeScript

Jan 04, 2017

Gets a script that updates the ConfigurationLogging Service database schema.

## Syntax

```
Get-LogDBVersionChangeScript -DatabaseName <String> -TargetVersion <Version> [-DataStore <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns a database script that can be used to upgrade or downgrade the site or secondary schema for the ConfigurationLogging Service from the current schema version to a different version.

## Related topics

[Get-LogInstalledDBVersion](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database instance to which the update applies.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TargetVersion**<Version>

Specifies the version of the database you want to update to.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-DataStore**<String>

Specifies the logical name of the data store for the ConfigurationLogging Service. Can be either be 'Site' or the logical name of the secondary data store.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | Site  |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

System.Management.Automation.PSObject

A PSObject containing the required SQL script for application to a database.

**Notes**

The PSObject returned by this cmdlet contains the following properties:

- Script The raw text of the SQL script to apply the update, or null in the case when no upgrade path to the specified target version exists.
- NeedExclusiveAccess Indicates whether all services in the service group must be shut down during the update or not.
- CanUndo Indicates whether the generated script allows the updated schema to be reverted to the state prior to the update.

Scripts to update the schema version are stored in the database so any service in the service group can obtain these scripts. Extreme caution should be exercised when using update scripts. Citrix recommends backing up the database before attempting to upgrade the schema. Database update scripts may require exclusive use of the schema and so may not be able to execute while any ConfigurationLogging services are running. However, this depends on the specific update being carried out.

After a schema update has been carried out, services that require the previous version of the schema may cease to operate. The ServiceState parameter reported by the Get-LogServiceStatus command provides information about service compatibility. For example, if the schema has been upgraded to a more recent version that a service cannot use, the service reports "DBNewerVersionThanService".

If the command fails, the following errors can be returned.

## Error Codes

-----

### NoOp

The operation was successful but had no effect.

### NoDBConnections

The database connection string for the ConfigurationLogging Service has not been specified.

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

### PermissionDenied

You do not have permission to execute this command.

### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

### CommunicationError

There was a problem communicating with the remote service.

### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> $update = Get-LogDBVersionChangeScript -DatabaseName MyDb -TargetVersion 1.0.75.0
```

```
C:\PS> $update.Script > update_75.sql
```

Gets an SQL update script to update the current schema to version 1.0.75.0. The resulting update\_75.sql script is suitable for direct use with the SQL Server SQLCMD utility.

# Get-LogHighLevelOperation

Jan 04, 2017

Gets high level operations

## Syntax

```
Get-LogHighLevelOperation [-Id <Guid>] [-Text <String>] [-StartTime <DateTime>] [-Source <String>] [-EndTime <DateTime>] [-IsSuccessful <Boolean>] [-User <String>] [-AdminMachineIP <String>] [-OperationType <OperationType>] [-TargetType <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves high level operations matching the specified criteria. If no parameters are specified this cmdlet returns all high level operations.

## Related topics

[Start-LogHighLevelOperation](#)

[Stop-LogHighLevelOperation](#)

[Get-LogLowLevelOperation](#)

## Parameters

**-Id**<Guid>

Gets the high level operation with the specified identifier.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Text**<String>

Gets high level operations with the specified text

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartTime**<DateTime>

Gets high level operations with the specified start time

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Source**<String>

Gets high level operations with the specified source.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-EndTime**<DateTime>

Gets high level operations with the specified end time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsSuccessful**<Boolean>

Gets high level operations with the specified success indicator.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-User**<String>

Gets high level operations logged by the specified user.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminMachineIP**<String>

Gets high level operations logged from the machine with the specified IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OperationType**<OperationType>

Gets high level operations with the specified operation type. Values can be:

- o AdminActivity - to get operations which log administration activity.
- o ConfigurationChange - to get operations which log configuration changes.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TargetType**<String>

Gets high level operations with the specified target type. The target type describes the type of object that was the target of the configuration change. For example, "Session" or "Machine".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

---

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Log\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Log\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

Citrix.ConfigurationLogging.Sdk.HighLevelOperation

The returned HighLevelOperation object has the following properties:

- o Id - The unique identifier of the operation.
- o Text - A description of the operation.



- o StartTime - The date and time that the operation started.
- o EndTime - The date and time that the operation completed. This will be null if the operation is still in progress, or if the operation never completed.
- o IsSuccessful - Indicates whether the operation completed successfully or not. This will be null if the operation is still in progress, or if the operation never completed.
- o User - The identifier of the administrator who performed the operation.
- o AdminMachineIP - The IP address of the machine that the operation was initiated from.
- o Source - Identifies the XenDesktop console, or custom script, where the operation was initiated from. For example, "Desktop Studio", "Desktop Director", "My custom script".
- o OperationType - The operation type. This can be 'AdminActivity' or 'ConfigurationChange'.
- o TargetTypes - Identifies the type of objects that were affected by the operation. For example, "Catalog" or "Desktop","Machine".
- o Parameters - The names and values of the parameters that were supplied for the operation.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Get-LogHighLevelOperation
Get all high level operations
```

### ----- EXAMPLE 2 -----

```
C:\PS> Get-LogHighLevelOperation -OperationType ConfigurationChange
Get high level operations which log configuration changes.
```

### ----- EXAMPLE 3 -----

```
C:\PS> Get-LogHighLevelOperation -OperationType AdminActivity
Get high level operations which log administration activities.
```

### ----- EXAMPLE 4 -----

```
C:\PS> Get-LogHighLevelOperation -Filter{ StartTime -ge "2013-02-27 09:00:00" -and EndTime -le "2013-02-27 18:00:00"}
Use advanced filtering to get high level operations with a start time on or after "2013-02-27 09:00:00" and an end time on or before "2013-02-27 18:00:00".
```

### ----- EXAMPLE 5 -----

```
C:\PS> Get-LogHighLevelOperation -EndTime $null
C:\PS> Get-LogHighLevelOperation -IsSuccessful $null
Either of these commands will get high level operations which have not yet been completed.
```

### ----- EXAMPLE 6 -----

```
C:\PS> Get-LogHighLevelOperation -User "DOMAIN\UserName"
```

Get high level operations performed by user "DOMAIN\UserName".

# Get-LogInstalledDBVersion

Jan 04, 2017

Gets a list of all available database schema versions for the ConfigurationLogging Service.

## Syntax

```
Get-LogInstalledDBVersion [-Upgrade] [-Downgrade] [-DataStore <String>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Returns the current version of the ConfigurationLogging Service database schema, if no flags are set, otherwise returns versions for which upgrade or downgrade scripts are available and have been stored in the database.

## Related topics

### Parameters

#### **-Upgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be updated should be returned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-Downgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be reverted should be returned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-DataStore**<String>

Specifies the database connection logical name the schema script should be returned for. The parameter is optional.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Return Values

System.Version

The Get-LogInstalledDbVersion command returns objects containing the new definition of the ConfigurationLogging Service database schema version.

Major <Integer>

Minor <Integer>

Build <Integer>

Revision <Integer>

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidParameterCombination

Both the Upgrade and Downgrade flags were specified.

NoOp

The operation was successful but had no effect.

NoDBConnections

The database connection string for the ConfigurationLogging Service has not been specified.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-LogInstalledDBVersion
```

```
Major Minor Build Revision
```

```
-----
```

```
5 6 0 0
```

Get the currently installed version of the ConfigurationLogging Service database schema.

----- **EXAMPLE 2** -----

```
c:\PS>Get-LogInstalledDBVersion -Upgrade
```

```
Major Minor Build Revision
```

```
-----
```

```
6 0 0 0
```

Get the versions of the ConfigurationLogging Service database schema for which upgrade scripts are supplied.

# Get-LogLowLevelOperation

Jan 04, 2017

Gets low level operations

## Syntax

```
Get-LogLowLevelOperation [-Id <Guid>] [-HighLevelOperationId <Guid>] [-StartTime <DateTime>] [-EndTime <DateTime>] [-IsSuccessful <Boolean>] [-User <String>] [-AdminMachineIP <String>] [-Text <String>] [-Source <String>] [-SourceSdk <String>] [-OperationType <OperationType>] [-TargetType <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves low level operations matching the specified criteria. If no parameters are specified this cmdlet returns all low level operations.

## Related topics

[Get-LogLowLevelOperation](#)

## Parameters

**-Id**<Guid>

Gets the low level operation with the specified identifier.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-HighLevelOperationId**<Guid>

Gets low level operations for the high level operation with the specified identifier.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartTime**<DateTime>

Gets low level operations with the specified start time

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-EndTime**<DateTime>

Gets low level operations with the specified end time.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsSuccessful**<Boolean>

Gets low level operations with the specified success indicator.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-User**<String>

Gets low level operations logged by the specified administrator.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminMachineIP**<String>

Gets low level operations logged from the machine with the specified IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Text**<String>

Gets low level operations with the specified text

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Source**<String>

Gets low level operations with the specified source.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-SourceSdk**<String>

Gets low level operations logged from the SDK with the specified identifier.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-OperationType**<OperationType>

Gets low level operations with the specified operation type. Values can be:

- o AdminActivity - to get operations which log administration activity.



o ConfigurationChange - to get operations which log configuration changes.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TargetType**<String>

Gets low level operations with the specified target type. The target type describes the type of object that was the target of the configuration change. For example, "Session" or "Machine".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Log\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Log\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.ConfigurationLogging.Sdk.LowLevelOperation

The returned LowLevelOperation object has the following properties:

- o Id - The unique identifier of the operation.
- o Text - A description of the operation.
- o HighLevelOperationId - The unique identifier of the related high level operation.
- o StartTime - The date and time that the operation started.
- o EndTime - The date and time that the operation completed. This will be null if the operation is still in progress, or if the operation never completed.
- o IsSuccessful - Indicates whether the operation completed successfully or not. This will be null if the operation is still in progress, or if the operation never completed.
- o AdminSid - The identifier of the administrator who performed the operation.
- o AdminMachineIP - The IP address of the machine that the operation was performed on.
- o Source - The name of the XenDesktop service that the operation was performed on; for example, "MachineCreation", "DelegatedAdmin".
- o SourceSdk - The identifier of the XenDesktop service SDK through which the operation was performed; for example, "Prov", "Admin".
- o OperationType - The operation type. This can be 'AdminActivity' or 'ConfigurationChange'.
- o TargetTypes - Identifies the type of objects that were affected by the operation. For example, "Catalog" or "Desktop","Machine".
- o Parameters - The names and values of the parameters that were supplied for the operation.
- o Details - A collection of OperationDetail objects containing specific information about each object affected by the operation.

Each OperationDetail object in the 'Details' collection has the following properties:

- o TargetUid - The unique identifier of the target object affected by the operation.
- o TargetName - The name of the target object affected by the operation.
- o TargetType - The type of the target object.
- o Text - The description of operation performed on the target object.
- o StartTime - The date and time that the operation started.
- o EndTime - The date and time that the operation completed. This will be null if the operation is still in progress, or if the operation never completed.
- o IsSuccessful - Indicates whether the operation completed successfully or not. This will be null if the operation is still in progress, or if the operation didn't complete.

The following properties will be set if the operation changed a property on the object:

- o PropertyName - The name of the changed property.
- o NewValue - The new property value.
- o PreviousValue - The previous property value.
- o AddValue - If the object property contains a set of values, this specifies the new value which was added to the set.
- o RemoveValue - If the object property contains a set of values, this specifies the value which was removed from the set.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Get-LogLowLevelOperation
Get all low level operations
```

### ----- EXAMPLE 2 -----

```
C:\PS> Get-LogLowLevelOperation -OperationType ConfigurationChange
Get low level operations which log configuration changes.
```

### ----- EXAMPLE 3 -----

```
C:\PS> Get-LogLowLevelOperation -OperationType AdminActivity
Get low level operations which log administration activities.
```

### ----- EXAMPLE 4 -----

```
C:\PS> Get-LogLowLevelOperation -Filter{ StartTime -ge "2013-02-27 09:00:00" -and EndTime -le "2013-02-27 18:00:00"}
Use advanced filtering to get low level operations with a start time on or after "2013-02-27 09:00:00" and an end time on or before "2013-02-27 18:00:00".
```

### ----- EXAMPLE 5 -----

```
C:\PS> Get-LogLowLevelOperation -EndTime $null
```

```
C:\PS> Get-LogLowLevelOperation -IsSuccessful $null
```

Either of these commands will get low level operations which have not yet been completed.

----- **EXAMPLE 6** -----

```
C:\PS> Get-LogLowLevelOperation -User "DOMAIN\UserName"
```

Get low level operations performed by user "DOMAIN\UserName".

# Get-LogService

Jan 04, 2017

Gets the service record entries for the ConfigurationLogging Service.

## Syntax

```
Get-LogService [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns instances of the ConfigurationLogging Service that the service publishes. The service records contain account security identifier information that can be used to remove each service from the database.

A database connection for the service is required to use this command.

## Related topics

### Parameters

#### **-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Log\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-Sort By**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

### **-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Log\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

Citrix.ConfigurationLogging.Sdk.Service

The Get-LogServiceInstance command returns an object containing the following properties.

Uid <Integer>

Specifies the unique identifier for the service in the group. The unique identifier is an index number.

ServiceHostId <Guid>

Specifies the unique identifier for the service instance.

DNSName <String>

Specifies the domain name of the host on which the service runs.

MachineName <String>

Specifies the short name of the host on which the service runs.

CurrentState <Citrix.Fma.Sdk.ServiceCore.ServiceState>

Specifies whether the service is running, started but inactive, stopped, or failed.

LastStartTime <DateTime>



Specifies the date and time at which the service was last restarted.

LastActivityTime <DateTime>

Specifies the date and time at which the service was last stopped or restarted.

OSType

Specifies the operating system installed on the host on which the service runs.

OSVersion

Specifies the version of the operating system installed on the host on which the service runs.

ServiceVersion

Specifies the version number of the service instance. The version number is a string that reflects the full build version of the service.

DatabaseUserName <string>

Specifies for the service instance the Active Directory account name with permissions to access the database. This will be either the machine account or, if the database is running on a controller, the NetworkService account.

Sid <string>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

ActiveSiteServices <string[]>

Specifies the names of active site services currently running in the service. Site services are components that perform long-running background processing in some services. This field is empty for services that do not contain site services.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### PartialData

Only a subset of the available data was returned.

#### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

#### CouldNotQueryDatabase

The query required to get the database was not defined.

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-LogService
```

```
Uid          : 1
ServiceHostId : aef6f464-f1ee-4042-a523-66982e0cecd0
DNSName      : MyServer.company.com
MachineName  : MYSERVER
CurrentState  : On
LastStartTime : 04/04/2011 15:25:38
LastActivityTime : 04/04/2011 15:33:39
OSType       : Win32NT
OSVersion    : 6.1.7600.0
ServiceVersion : 5.1.0.0
DatabaseUserName : NT AUTHORITY\NETWORK SERVICE
SID          : S-1-5-21-2316621082-1546847349-2782505528-1165
ActiveSiteServices : {MySiteService1, MySiteService2...}
Get all the instances of the ConfigurationLogging Service running in the current service group.
```

# Get-LogServiceAddedCapability

Jan 04, 2017

Gets any added capabilities for the ConfigurationLogging Service on the controller.

## Syntax

```
Get-LogServiceAddedCapability [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables updates to the ConfigurationLogging Service on the controller to be detected.

You do not need to configure a database connection before using this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.String

String containing added capabilities.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- **EXAMPLE 1** -----

```
c:\PS>Get-LogServiceAddedCapability
```

Get the added capabilities of the ConfigurationLogging Service.

# Get-LogServiceInstance

Jan 04, 2017

Gets the service instance entries for the ConfigurationLogging Service.

## Syntax

```
Get-LogServiceInstance [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns service interfaces published by the instance of the ConfigurationLogging Service. Each instance of a service publishes multiple interfaces with distinct interface types, and each of these interfaces is represented as a ServiceInstance object. Service instances can be used to register the service with a central configuration service so that other services can use the functionality.

You do not need to configure a database connection to use this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.ConfigurationLogging.Sdk.ServiceInstance

The Get-LogServiceInstance command returns an object containing the following properties.

ServiceGroupUid <Guid>

Specifies the unique identifier for the service group of which the service is a member.

ServiceGroupName <String>

Specifies the name of the service group of which the service is a member.

ServiceInstanceUID <Guid>

Specifies the unique identifier for registered service instances, which are service instances held by and obtained from a

central configuration service. Unregistered service instances do not have unique identifiers.

ServiceType <String>

Specifies the service instance type. For this service, the service instance type is always Log.

Address

Specifies the address of the service instance. The address can be used to access the service and, when registered in the central configuration service, can be used by other services to access the service.

Binding

Specifies the binding type that must be used to communicate with the service instance. In this release of XenDesktop, the binding type is always 'wcf\_HTTP\_kerb'. This indicates that the service provides a Windows Communication Foundation endpoint that uses HTTP binding with integrated authentication.

Version

Specifies the version of the service instance. The version number is used to ensure that the correct versions of the services are used for communications.

ServiceAccount <String>

Specifies the Active Directory account name for the machine on which the service instance is running. The account name is used to provide information about the permissions required for interservice communications.

ServiceAccountSid <String>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

InterfaceType <String>

Specifies the interface type. Each service can provide multiple service instances, each for a different purpose, and the interface defines the purpose. Available interfaces are:

SDK - for PowerShell operations

InterService - for operations between different services

Peer - for communications between services of the same type

Metadata <Citrix.ConfigurationLogging.Sdk.Metadata[]>

The collection of metadata associated with registered service instances, which are service instances held by and obtained from a central configuration service. Metadata is not stored for unregistered service instances.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-LogServiceInstance
```

```
Address      : http://MyServer.com:80/Citrix/ConfigurationLoggingService
Binding      : wcf_HTTP_kerb
InterfaceType : SDK
Metadata     :
MetadataMap  :
ServiceAccount : ENG\MyAccount$
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164
ServiceGroupName : MyServiceGroup
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000
ServiceType   : Log
Version      : 1
```

```
Address      : http://MyServer.com:80/Citrix/ConfigurationLoggingService/IServiceApi
Binding      : wcf_HTTP_kerb
InterfaceType : InterService
Metadata     :
MetadataMap  :
```

ServiceAccount : ENG\MyAccount  
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164  
ServiceGroupName : MyServiceGroup  
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d  
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000  
ServiceType : Log  
Version : 1

Get all instances of the ConfigurationLogging Service running on the specified machine. For remote services, use the AdminAddress parameter to define the service for which the interfaces are required. If the AdminAddress parameter has not been specified for the runspace, service instances running on the local machine are returned.



# Get-LogServiceStatus

Jan 04, 2017

Gets the current status of the ConfigurationLogging Service on the controller.

## Syntax

```
Get-LogServiceStatus [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables the status of the ConfigurationLogging Service on the controller to be monitored. If the service has multiple data stores it will return the overall state as an aggregate of all the data store states. For example, if the site data store status is OK and the secondary data store status is DBUnconfigured then it will return DBUnconfigured.

## Related topics

[Get-LogDataStore](#)

[Set-LogDBConnection](#)

[Test-LogDBConnection](#)

[Get-LogDBConnection](#)

[Get-LogDBSchema](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Get-LogServiceStatus command returns an object containing the status of the ConfigurationLogging Service together with extra diagnostics information.

DBUnconfigured

The ConfigurationLogging Service does not have a database connection configured.

#### DBRejectedConnection

The database rejected the logon attempt from the ConfigurationLogging Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

#### InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the ConfigurationLogging Service schema has not been added to the database.

#### DBNotFound

The specified database could not be located with the configured connection string.

#### DBNewerVersionThanService

The version of the ConfigurationLogging Service currently in use is incompatible with the version of the ConfigurationLogging Service schema on the database. Upgrade the ConfigurationLogging Service to a more recent version.

#### DBOlderVersionThanService

The version of the ConfigurationLogging Service schema on the database is incompatible with the version of the ConfigurationLogging Service currently in use. Upgrade the database schema to a more recent version.

#### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

#### OK

The ConfigurationLogging Service is running and is connected to a database containing a valid schema.

#### Failed

The ConfigurationLogging Service has failed.

#### Unknown

(0) The service status cannot be determined.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-LogServiceStatus
```

## DBUnconfigured

Get the current status of the ConfigurationLogging Service.

# Get-LogSite

Jan 04, 2017

Gets global configuration logging settings.

## Syntax

```
Get-LogSite [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet retrieves the global configuration logging settings.

## Related topics

[Set-LogSite](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.ConfigurationLogging.Sdk.Site

Get-LogSite returns an object containing the following properties:

- o Locale - the current language that logs should be recorded in. Can be: 'en', 'ja', 'zh-CN', 'de', 'es' or 'fr'.
- o State - the current state of configuration logging. Can be: Enabled, Disabled, Mandatory or NotSupported.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-LogSite  
Gets configuration logging site settings.
```

# Get-LogSummary

Jan 04, 2017

Gets operations logged within time intervals inside a date range.

## Syntax

```
Get-LogSummary [-StartDateRange <DateTime>] [-EndDateRange <DateTime>] [-IntervalSeconds <Int64>] [-GetLowLevelOperations] [-IncludeIncomplete] [-OperationType <OperationType>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Get-LogSummary cmdlet retrieves summary counts of operations logged within time intervals inside a date range. The returned data indicates the rate at which configuration changes and activities were performed out within a time period.

## Related topics

[Get-LogHighLevelOperation](#)

[Get-LogLowLevelOperation](#)

## Parameters

**-StartDateRange**<DateTime>

Specifies the start of the summary period date range

|                        |                     |
|------------------------|---------------------|
| Required?              | false               |
| Default Value          | 1900-01-01 00:00:00 |
| Accept Pipeline Input? | false               |

**-EndDateRange**<DateTime>

Specifies the end of the summary period date range.

|                        |                 |
|------------------------|-----------------|
| Required?              | false           |
| Default Value          | DateTime.UtcNow |
| Accept Pipeline Input? | false           |

**-IntervalSeconds**<Int64>

Specifies the size, in seconds, of each time interval required within the summary date range. If this is not specified, is null, zero or exceeds the specified date range, it defaults to the total number of seconds between EndDateRange and StartDateRange.

|           |       |
|-----------|-------|
| Required? | false |
|           |       |

|                        |                                                                           |
|------------------------|---------------------------------------------------------------------------|
| Default Value          | Total number of seconds in the EndDateRange and StartDateRange time span. |
| Accept Pipeline Input? | false                                                                     |

**-GetLowLevelOperations**<SwitchParameter>

Specifies if the cmdlet should return low level operation summary counts.

|                        |                                                      |
|------------------------|------------------------------------------------------|
| Required?              | false                                                |
| Default Value          | \$false - high level operations counts are returned. |
| Accept Pipeline Input? | false                                                |

**-IncludeIncomplete**<SwitchParameter>

Specifies if incomplete operations should be included in the returned summary counts.

|                        |                                               |
|------------------------|-----------------------------------------------|
| Required?              | false                                         |
| Default Value          | \$false - incomplete operations are excluded. |
| Accept Pipeline Input? | false                                         |

**-OperationType**<OperationType>

Specifies the type of logged operations to include. Values can be 'AdminActivity' or 'ConfigurationChange'

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.Collections.Generic.Dictionary<string, int>

The summary data is returned as a collection of dictionary items. The 'Key' value of each dictionary item specifies the start of the time interval within the overall summary date range. The 'Value' data in each dictionary item contains the count of operations which were started within that time interval.

## Notes

If the specified summary date range and interval period will result in more than 50,000 intervals being returned Get-LogSummary will generate an error and abort the operation.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> $logSummary = Get-LogSummary
```

Get a summary of all completed high level operations. The returned log summary collection will contain a single item for the time period spanning the entire [1900-01-01 00:00:00]-[UtcNow] date range. e.g.

Key ==> Value

```
01/01/1900 00:00:00 ==> 41
```

----- **EXAMPLE 2** -----

```
C:\PS> $daily = 60*60*24
```

```
C:\PS> [DateTime]$startRange = "2013-02-01 14:50:39"
```

```
C:\PS> [DateTime]$endRange = $startRange.AddDays(14)
```

```
C:\PS> Get-LogSummary -StartDateRange $startRange -EndDateRange $endRange -intervalSeconds $daily
```

Gets a summarised count of completed high level operations logged over two weeks, at daily intervals. The returned log summary collection will contain multiple items; one for each day in the summary date range. - e.g.

Key ==> Value

```
01/02/2013 14:50:39 ==> 0
```

```
02/02/2013 14:50:39 ==> 4
```

```
03/02/2013 14:50:39 ==> 21
```

```
04/02/2013 14:50:39 ==> 0
```

```
05/02/2013 14:50:39 ==> 0
```

```
06/02/2013 14:50:39 ==> 0
```

```
07/02/2013 14:50:39 ==> 5
```

```
08/02/2013 14:50:39 ==> 0
```

```
09/02/2013 14:50:39 ==> 0
```

```
10/02/2013 14:50:39 ==> 0
```

```
11/02/2013 14:50:39 ==> 0
```

```
12/02/2013 14:50:39 ==> 7
```

```
13/02/2013 14:50:39 ==> 0
14/02/2013 14:50:39 ==> 0
15/02/2013 14:50:39 ==> 12
```

----- **EXAMPLE 3** -----

```
C:\PS> $hourly = 60*60
C:\PS> [DateTime]$startRange = "2013-02-03 00:00:00"
C:\PS> [DateTime]$endRange = "2013-02-03 23:59:59"
C:\PS> Get-LogSummary -StartDateRange $startRange -EndDateRange $endRange -intervalSeconds $hourly -GetLowLevelOperations
```

Gets a summarised count of completed low level operations logged during a day, at hourly intervals. The returned log summary collection will contain multiple items; one for each hour in the summary date range - e.g.

```
Key ==> Value

04/03/2013 00:00:00 ==> 12
04/03/2013 01:00:00 ==> 10
04/03/2013 02:00:00 ==> 5
.
.
.
04/03/2013 21:00:00 ==> 26
04/03/2013 22:00:00 ==> 0
04/03/2013 23:00:00 ==> 9
```



# Remove-LogOperation

Jan 04, 2017

Deletes configuration logs

## Syntax

```
Remove-LogOperation -DatabaseCredentials <PSCredential> [-StartDateRange <DateTime>] [-EndDateRange <DateTime>] [-IncludeIncomplete] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-LogOperation -UserName <String> [-Password <String>] [-StartDateRange <DateTime>] [-EndDateRange <DateTime>] [-IncludeIncomplete] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-LogOperation -UserName <String> -SecurePassword <SecureString> [-StartDateRange <DateTime>] [-EndDateRange <DateTime>] [-IncludeIncomplete] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Remove-LogOperation deletes logs from the Configuration Logging database. This cmdlet targets high level operation logs for deletion. The associated low level operations logs are deleted as part of this operation via cascade deletion functionality present in the configuration logging database schema.

## Related topics

### Parameters

#### **-DatabaseCredentials**<PSCredential>

Specifies the credentials of a database user with permission to delete records from the Configuration Logging database.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-UserName**<String>

Specifies the user name of a database user with permission to delete records from the Configuration Logging database.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-SecurePassword**<SecureString>

Specifies the password a database user, in secure string form, with permission to delete records from the Configuration Logging database.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-Password**<String>

Specifies the password of a database user with permission to delete records from the Configuration Logging database.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartDateRange**<DateTime>

Specifies the start time of the earliest high level operation to delete

|                        |              |
|------------------------|--------------|
| Required?              | false        |
| Default Value          | DateTime.Min |
| Accept Pipeline Input? | false        |

**-EndDateRange**<DateTime>

Specifies the end time of the latest high level operation to delete

|                        |                 |
|------------------------|-----------------|
| Required?              | false           |
| Default Value          | DateTime.UtcNow |
| Accept Pipeline Input? | false           |

**-IncludeIncomplete**<SwitchParameter>

Specifies if incomplete high level operations should be deleted.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Remove-LogOperation -UserName "DOMAIN\User" -Password "UserPassword"
Remove all completed operation logs
```

### ----- EXAMPLE 2 -----

```
C:\PS> Remove-LogOperation -UserName "DOMAIN\User" -Password "UserPassword" -IncludeIncomplete
Remove all operations
```

### ----- EXAMPLE 3 -----

```
C:\PS> Remove-LogOperation -username "domain\userName" -password "password" -StartDateRange "2013-01-01 12:00:00" -EndDateRange "2013-01-31 12:00:00"
Delete logs started on or after "2013-01-01 12:00:00" and completed on or before "2013-01-31 12:00:00"
```

### ----- EXAMPLE 4 -----

```
C:\PS> $securePassword = ConvertTo-SecureString -String "UserPassword" -AsPlainText -Force
C:\PS> $credentials = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList "DOMAIN\UserName", $securePassword
C:\PS> Remove-LogOperation -StartDateRange -DatabaseCredentials $credentials
Delete logs by supplying database user credentials via a credentials object.
```

# Remove-LogServiceMetadata

Jan 04, 2017

Removes metadata from the given Service.

## Syntax

```
Remove-LogServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-LogServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-LogServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-LogServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Service.

## Related topics

[Set-LogServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Service[]>

Objects to which metadata is to be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-LogService | % { Remove-LogServiceMetadata -Map $_.MetadataMap }  
Remove all metadata from all Service objects.
```

# Remove-LogSiteMetadata

Jan 04, 2017

Removes metadata from the given Site.

## Syntax

```
Remove-LogSiteMetadata -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-LogSiteMetadata -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-LogSiteMetadata -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-LogSiteMetadata -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Site.

## Related topics

[Set-LogSiteMetadata](#)

## Parameters

**-Name**<String>

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |



### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-LogSite | % { Remove-LogSiteMetadata -Map $_.MetadataMap }  
Remove all metadata from all Site objects.
```

# Reset-LogDataStore

Jan 04, 2017

Refreshes the database string currently being used by the Log service.

## Syntax

```
Reset-LogDataStore [-DataStore] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the string for the database connection currently being used by the ConfigurationLogging Service. Can only be called for secondary data stores.

There is no requirement for a database connection to be configured in order for this command to be used.

## Related topics

[Get-LogDataStore](#)

## Parameters

### **-DataStore**<String>

Specifies the database connection logical name to be used by the ConfigurationLogging Service. Can be either be 'Site' or the logical name of the secondary data store. Specifying the site data store will display an error because this operation is not supported for site data stores.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          | Site  |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.ConfigurationLogging.Sdk.ServiceStatus

The status of the specified data store.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

### PermissionDenied

You do not have permission to execute this command.

### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

### CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Reset-LogDataStore -DataStore Secondary
```

```
OK
```

Refresh the database connection string for the ConfigurationLogging Service.

# Reset-LogServiceGroupMembership

Jan 04, 2017

Reloads the access permissions and configuration service locations for the ConfigurationLogging Service.

## Syntax

```
Reset-LogServiceGroupMembership [-ConfigServiceInstance] <ServiceInstance[]> [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

## Detailed Description

Enables you to reload ConfigurationLogging Service access permissions and configuration service locations. The Reset-LogServiceGroupMembership command must be run on at least one instance of the service type (Log) after installation and registration with the configuration service. Without this operation, the ConfigurationLogging services will be unable to communicate with other services in the XenDesktop deployment. When the command is run, the services are updated when additional services are added to the deployment, provided that the configuration service is not stopped. The Reset-LogServiceGroupMembership command can be run again to refresh this information if automatic updates do not occur when new services are added to the deployment. If more than one configuration service instance is passed to the command, the first instance that meets the expected service type requirements is used.

## Related topics

### Parameters

**-ConfigServiceInstance**<ServiceInstance[]>

Specifies the configuration service instance object that represents the service instance for the type 'InterService' that references a configuration service for the deployment.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? |       |

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Input Type

Citrix.ConfigurationLogging.Sdk.ServiceInstance[] Service instances containing a ServiceInstance object that refers to the central configuration service interservice interface can be piped to the Reset-LogServiceGroupMembership command.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoSuitableServiceInstance

None of the supplied service instance objects were suitable for resetting service group membership.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-LogServiceGroupMembership
```

Reset the service group membership for a service in a deployment where the configuration service is configured and running on the same machine as the service.

### ----- EXAMPLE 2 -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config -AdminAddress OtherServer.example.com | Reset-LogServiceGroupmembership
```

Reset the service group membership for a service in a deployment where the configuration service that is configured and running on a machine named 'OtherServer.example.com'.

---



# Set-LogDBConnection

Jan 04, 2017

Configures a database connection for the ConfigurationLogging Service.

## Syntax

```
Set-LogDBConnection [-DBConnection] <String> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [[-DataStore] <String>]  
[<CommonParameters>]
```

## Detailed Description

Configures a connection to a database in which the ConfigurationLogging Service can store its state. The service will attempt to connect and start using the database immediately after the connection is configured. The database connection string is updated to the specified value regardless of whether it is valid or not. Specifying an invalid connection string prevents a service from functioning until the error is corrected.

After a connection is configured, you cannot alter it without first clearing it (by setting the connection to \$null).

You do not need to configure a database connection to use this command.

## Related topics

[Get-LogServiceStatus](#)

[Get-LogDBConnection](#)

[Test-LogDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be used by the ConfigurationLogging Service. Passing in \$null will clear any existing database connection configured.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Force**<SwitchParameter>

If present, allows the local administrator to set the connection string to null when there are problems contacting the database or other services.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level

operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**-DataStore**<String>

Specifies the logical name of the data store for the ConfigurationLogging Service. Can be either be 'Site' or the logical name of the secondary data store.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | Site  |
| Accept Pipeline Input? | false |

**Return Values**

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Set-LogDBConnection command returns an object containing the status of the ConfigurationLogging Service together with extra diagnostics information.

DBUnconfigured

The ConfigurationLogging Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the ConfigurationLogging Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the ConfigurationLogging Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

#### DBNewerVersionThanService

The version of the ConfigurationLogging Service currently in use is incompatible with the version of the ConfigurationLogging Service schema on the database. Upgrade the ConfigurationLogging Service to a more recent version.

#### DBOlderVersionThanService

The version of the ConfigurationLogging Service schema on the database is incompatible with the version of the ConfigurationLogging Service currently in use. Upgrade the database schema to a more recent version.

#### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

#### OK

The ConfigurationLogging Service is running and is connected to a database containing a valid schema.

#### Failed

The ConfigurationLogging Service has failed.

#### Unknown

The status of the ConfigurationLogging Service cannot be determined.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### InvalidDBConnectionString

The database connection string has an invalid format.

#### DatabaseConnectionDetailsAlreadyConfigured

There was already a database connection configured. After a configuration is set, it can only be set to \$null.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-LogDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Configures a database connection string for the ConfigurationLogging Service.

----- **EXAMPLE 2** -----

```
c:\PS>Set-LogDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Configures an invalid database connection string for the ConfigurationLogging Service.

# Set-LogServiceMetadata

Jan 04, 2017

Adds or updates metadata on the given Service.

## Syntax

```
Set-LogServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Set-LogServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

```
Set-LogServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

```
Set-LogServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

## Detailed Description

Allows you to store additional custom data against given Service objects.

## Related topics

[Remove-LogServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Service[]>

Objects to which metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

### **-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with `@{"name1" = "val1"; "name2" = "val2"}`) or a string dictionary (created with `new-object "System.Collections.Generic.Dictionary[String,String]"`).

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

### **-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Service specified. The property cannot contain any of the following characters `\;/#.*?=<>|[]()`

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the `Start-LogHighLevelOperation` and `Stop-LogHighLevelOperation` cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

System.Collections.Generic.Dictionary[String,String]

Set-LogServiceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
c:\PS>Set-LogServiceMetadata -ServiceHostId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Key      | Value |
|----------|-------|
| ---      | ----- |
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the Service with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.



# Set-LogSite

Jan 04, 2017

Sets global configuration logging settings.

## Syntax

```
Set-LogSite [-State <LoggingState>] [-Locale <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet sets the global configuration logging settings.

## Related topics

[Get-LogSite](#)

## Parameters

**-State**<LoggingState>

Sets the state of configuration logging. Values can be:

o Enabled - state changes will be logged. If logging is unavailable, the state change will still be permitted.  
o Disabled - state changes will not be logged.

o Mandatory - state change will be logged. If logging is unavailable, the state change will not be permitted.

When the state is set to Enabled or Mandatory, XenDesktop services will attempt to log operations (which perform configuration changes) before performing them.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Locale**<String>

Sets the language that logs should be recorded in. Values can be: 'en', 'ja', 'zh-CN', 'de', 'es' or 'fr'.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-PassThru**<SwitchParameter>

Returns the affected record. By default, this cmdlet does not generate any output.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

Citrix.ConfigurationLogging.Sdk.Site

Current logging settings

### Notes

Configuration logging will automatically transition to a 'NotSupported' state if the logging feature is not licensed. Set-LogSite will reject request to set logging to 'Enabled' or 'Mandatory' while the logging feature is not licensed.

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Set-LogSite -Locale "zh-CN"
```

Set the logging language to Chinese. The logging state will be unchanged.

----- **EXAMPLE 2** -----

```
C:\PS> Set-LogSite -State "Mandatory"
```

Set the logging state to mandatory. The logging locale will be unaffected. In this state, no configuration change will be allowed unless it is successfully logged.

----- **EXAMPLE 3** -----

```
C:\PS> Set-LogSite -Locale "de" -State "Enabled"
```

Set the logging language to German and the state to Enabled.

# Set-LogSiteMetadata

Jan 04, 2017

Adds or updates metadata on the given Site.

## Syntax

```
Set-LogSiteMetadata -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-LogSiteMetadata -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-LogSiteMetadata -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-LogSiteMetadata -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given Site objects.

## Related topics

[Remove-LogSiteMetadata](#)

## Parameters

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Site specified. The property cannot contain any of the following characters \/:#.\*?=<> | []()"

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-LogSiteMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

### Examples

#### ----- EXAMPLE 1 -----

```
c:\PS>Set-LogSiteMetadata -SiteDUMMY_IdProperty 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Key      | Value |
|----------|-------|
| ---      | ----  |
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the Site with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Start-LogHighLevelOperation

Jan 04, 2017

Logs the start of a high level operation.

## Syntax

```
Start-LogHighLevelOperation -Text <String> -Source <String> [-StartTime <DateTime>] [-OperationType <OperationType>] [-TargetTypes <String[]>] [-Parameters <Hashtable>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Start-LogHighLevelOperation creates a log entry to record the start of a high level operation.

Start-LogHighLevelOperation can be called to log high level configuration changes which originate from customized configuration scripts. Start-LogHighLevelOperation should be called before the script invokes service SDK cmdlets which execute the configuration changes.

Start-LogHighLevelOperation returns a HighLevelOperation object which contains information about the started high level operation. The "Id" property of the returned HighLevelOperation uniquely identifies the started high level operation. This can be supplied into the "-LoggingId" parameter which is implemented in all service SDK cmdlets which execute loggable configuration changes.

High level operation logs are automatically created by the XenDesktop consoles when:

- o Initiating an operation which performs configuration changes.
- o Initiating an operation which performs an administration activity.

## Configuration Change and Administrator Activity

-----  
A configuration change is an operation which alters the configuration of the XenDesktop site. Examples of a configuration changes include:

- o Creating or editing a host.
- o Creating or editing a catalog.
- o Adding a user to a delivery group.
- o Deleting a machine.

An administrator activity operation doesn't directly alter site configuration, but it could be an operation carried out by an administrator as part of site management or helpdesk support. Examples of administrator activities include:

- o Shutdown/start/restart of a user desktop.
- o Studio or Director sending a message to a user.
- o Rebooting a user's desktop.

Once the change being logged has completed (whether successfully or not) the Stop-LogHighLevelOperation cmdlet should be called to log the completion of a started high level operation.

## Related topics

[Stop-LogHighLevelOperation](#)

[Get-LogHighLevelOperation](#)

## Parameters

**-Text**<String>

Specifies text to describe the high level operation.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Source**<String>

Specifies the source of the high level operation.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-StartTime**<DateTime>

Specifies the start time of the high level operation.

|                        |                 |
|------------------------|-----------------|
| Required?              | false           |
| Default Value          | DateTime.UtcNow |
| Accept Pipeline Input? | false           |

**-OperationType**<OperationType>

Specifies the type of operation being logged. Values can be:

- o AdminActivity - If the operation being logged performs an administration activity.
- o ConfigurationChange - If the operation being logged performs a configuration change.

|                        |                     |
|------------------------|---------------------|
| Required?              | false               |
| Default Value          | ConfigurationChange |
| Accept Pipeline Input? | false               |

**-TargetTypes**<String[]>

Specifies the names of the object types that will be affected by the operation being logged.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Parameters**<Hashtable>

Specifies the names and values of parameters that are supplied to the operation being logged.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |



## Return Values

Citrix.ConfigurationLogging.Sdk.HighLevelOperation

The newly logged high level operation start.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $succeeded = $false #indicates if high level operation succeeded.
C:\PS> # Log high level operation start.
C:\PS> $highLevelOp = Start-LogHighLevelOperation -Text "Create catalog" -Source "My Custom Script"
C:\PS>
C:\PS> try
C:\PS> {
C:\PS> # Create catalog object
C:\PS> $catalog = New-BrokerCatalog -Name "MyCatalog" -ProvisioningType Manual -AllocationType Permanent -MinimumFunctionalLevel 'LMAX' -LoggingId $highLevelOp.Id
C:\PS>
C:\PS> # Add a machine to the catalog
C:\PS> $machine = New-BrokerMachine -CatalogUid $catalog.Uid -MachineName "DOMAIN\Machine" -LoggingId $highLevelOp.Id
C:\PS> $succeeded = $true
C:\PS> }
C:\PS> catch{ "Error encountered" }
C:\PS>
C:\PS> finally{
C:\PS> # Log high level operation stop, and indicate its success
C:\PS> Stop-LogHighLevelOperation -HighLevelOperationId $highLevelOp.Id -IsSuccessful $succeeded
C:\PS> }
```

Creates an unmanaged catalog and assigns a machine to it, within the scope of a high level operation start and stop. The identifier of the high level operation is passed into the "-LoggingId" parameter of the service SDK cmdlets. The execution of the cmdlets in the services will create the low level operation logs for the supplied high level operation.

# Stop-LogHighLevelOperation

Jan 04, 2017

Logs the completion of a previously started high level operation.

## Syntax

Stop-LogHighLevelOperation -HighLevelOperationId <String> -IsSuccessful <Boolean> [-EndTime <DateTime>] [-AdminAddress <String>] [<CommonParameters>]

## Detailed Description

Stop-LogHighLevelOperation logs the completion of a started high level operation.

## Related topics

[Start-LogHighLevelOperation](#)

[Get-LogHighLevelOperation](#)

## Parameters

**-HighLevelOperationId**<String>

Specifies the identifier of the high level operation being completed.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-IsSuccessful**<Boolean>

Specifies if the started high level operation completed successfully.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-EndTime**<DateTime>

Specifies the end time of the high level operation being completed.

|                        |                  |
|------------------------|------------------|
| Required?              | false            |
| Default Value          | DateTime.UtcNow. |
| Accept Pipeline Input? | false            |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Examples

----- **EXAMPLE 1** -----

```

C:\PS> $succeeded = $false #indicates if high level operation succeeded.
C:\PS> # Log high level operation start.
C:\PS> $highLevelOp = Start-LogHighLevelOperation -Text "Create unmanaged catalog" -Source "My Custom Script"
C:\PS>
C:\PS> try
C:\PS> {
C:\PS> # Create catalog object
C:\PS> $catalog = New-BrokerCatalog -Name "MyCatalog" -ProvisioningType Manual -AllocationType Permanent -MinimumFunctionalLevel 'LMAX' -LoggingId $highLevelOp.Id
C:\PS>
C:\PS> # Add a machine to the catalog
C:\PS> $machine = New-BrokerMachine -CatalogUid $catalog.Uid -MachineName "DOMAIN\Machine" -LoggingId $highLevelOp.Id
C:\PS> $succeeded = $true
C:\PS> }
C:\PS> catch{ "Error encountered" }
C:\PS>
C:\PS> finally{
C:\PS> # Log high level operation stop, and indicate its success
C:\PS> Stop-LogHighLevelOperation -HighLevelOperationId $highLevelOp.Id -IsSuccessful $succeeded
C:\PS> }

```

Creates an unmanaged catalog and assigns a machine to it, within the scope of a high level operation start and stop. The identifier of the high level operation is passed into the "-LoggingId" parameter of the service SDK cmdlets. The execution of the cmdlets in the services will create the low level operation logs for the supplied high level operation.

# Test-LogDBConnection

Jan 04, 2017

Tests a database connection for the ConfigurationLogging Service.

## Syntax

```
Test-LogDBConnection [-DBConnection] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [[-DataStore] <String>] [<CommonParameters>]
```

## Detailed Description

Tests a connection to the database in which the ConfigurationLogging Service can store its state. The service will attempt to connect to the database without affecting the current connection to the database.

You do not have to clear the connection to use this command.

## Related topics

[Get-LogServiceStatus](#)

[Get-LogDBConnection](#)

[Set-LogDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be tested by the ConfigurationLogging Service.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? |       |

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**-DataStore<String>**

Specifies the logical name of the data store for the ConfigurationLogging Service. Can be either be 'Site' or the logical name of the secondary data store.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | Site  |
| Accept Pipeline Input? | false |

**Return Values**

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Test-LogDBConnection command returns an object containing the status of the ConfigurationLogging Service if the connection string of the specified data store were to be set to the string being tested, together with extra diagnostics information for the specified connection string.

DBRejectedConnection

The database rejected the logon attempt from the ConfigurationLogging Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the ConfigurationLogging Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the ConfigurationLogging Service currently in use is incompatible with the version of the ConfigurationLogging Service schema on the database. Upgrade the ConfigurationLogging Service to a more recent version.

DBOlderVersionThanService

The version of the ConfigurationLogging Service schema on the database is incompatible with the version of the ConfigurationLogging Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The Set-LogDBConnection command would succeed if it were executed with the supplied connection string.

Failed

The ConfigurationLogging Service has failed.

Unknown

The status of the ConfigurationLogging Service cannot be determined.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidDBConnectionString

The database connection string has an invalid format.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Test-LogDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Tests a database connection string for the ConfigurationLogging Service.

----- **EXAMPLE 2** -----

```
c:\PS>Test-LogDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Tests an invalid database connection string for the ConfigurationLogging Service.

# Citrix.DelegatedAdmin.Admin.V1

Jan 04, 2017

## Overview

| Name                                      | Description                                                      |
|-------------------------------------------|------------------------------------------------------------------|
| <a href="#">AdminDelegatedAdminSnapin</a> | The Delegated Administration Service PowerShell snap-in provides |
| <a href="#">Admin Filtering</a>           | Describes the common filtering options for XenDesktop cmdlets.   |

## Cmdlets

| Name                                            | Description                                                                                         |
|-------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <a href="#">Add-AdminPermission</a>             | Add permissions to the set of permissions of a role.                                                |
| <a href="#">Add-AdminRight</a>                  | Grants a given right to the specified administrator.                                                |
| <a href="#">Get-AdminAdministrator</a>          | Gets administrators configured for this site.                                                       |
| <a href="#">Get-AdminDBConnection</a>           | Gets the database string for the specified data store used by the DelegatedAdmin Service.           |
| <a href="#">Get-AdminDBSchema</a>               | Gets a script that creates the DelegatedAdmin Service database schema for the specified data store. |
| <a href="#">Get-AdminDBVersionChangeScript</a>  | Gets a script that updates the DelegatedAdmin Service database schema.                              |
| <a href="#">Get-AdminEffectiveAdministrator</a> | Retrieve the effective administrator objects for a user.                                            |
| <a href="#">Get-AdminEffectiveRight</a>         | Gets the set of Right objects associated with the current user.                                     |
| <a href="#">Get-AdminInstalledDBVersion</a>     | Gets a list of all available database schema versions for the DelegatedAdmin Service.               |
| <a href="#">Get-AdminPermission</a>             | Gets permissions configured for the site.                                                           |
| <a href="#">Get-AdminPermissionGroup</a>        | Gets permission groups configured for the site.                                                     |
| <a href="#">Get-AdminRevision</a>               | Gets the current revision of the delegated administration configuration data.                       |

| Get-AdminRole<br>Name             | Description                                                                   |
|-----------------------------------|-------------------------------------------------------------------------------|
| Get-AdminRoleConfiguration        | Gets role configurations for this site.                                       |
| Get-AdminScope                    | Gets scopes configured for this site.                                         |
| Get-AdminService                  | Gets the service record entries for the DelegatedAdmin Service.               |
| Get-AdminServiceAddedCapability   | Gets any added capabilities for the DelegatedAdmin Service on the controller. |
| Get-AdminServiceInstance          | Gets the service instance entries for the DelegatedAdmin Service.             |
| Get-AdminServiceStatus            | Gets the current status of the DelegatedAdmin Service on the controller.      |
| Import-AdminRoleConfiguration     | Imports role configuration data into the Delegated Administration Service.    |
| New-AdminAdministrator            | Adds a new administrator to the site.                                         |
| New-AdminRole                     | Adds a new custom role to the site.                                           |
| New-AdminScope                    | Adds a new scope to the site.                                                 |
| Remove-AdminAdministrator         | Removes administrators from the site.                                         |
| Remove-AdminAdministratorMetadata | Removes metadata from the given Administrator.                                |
| Remove-AdminPermission            | Remove permissions from the set of permissions of a role.                     |
| Remove-AdminRight                 | Removes rights from an administrator.                                         |
| Remove-AdminRole                  | Removes a role from the site.                                                 |
| Remove-AdminRoleMetadata          | Removes metadata from the given Role.                                         |
| Remove-AdminScope                 | Removes a scope from the site.                                                |
| Remove-AdminScopeMetadata         | Removes metadata from the given Scope.                                        |



| Remove-AdminServiceMetadata<br>Name | Description                                                                                        |
|-------------------------------------|----------------------------------------------------------------------------------------------------|
| Rename-AdminRole                    | Rename a role                                                                                      |
| Rename-AdminScope                   | Rename a scope                                                                                     |
| Reset-AdminServiceGroupMembership   | Reloads the access permissions and configuration service locations for the DelegatedAdmin Service. |
| Set-AdminAdministrator              | Sets the properties of an administrator.                                                           |
| Set-AdminAdministratorMetadata      | Adds or updates metadata on the given Administrator.                                               |
| Set-AdminDBConnection               | Configures a database connection for the DelegatedAdmin Service.                                   |
| Set-AdminRole                       | Set the properties of a role.                                                                      |
| Set-AdminRoleMetadata               | Adds or updates metadata on the given Role.                                                        |
| Set-AdminScope                      | Set the properties of a scope.                                                                     |
| Set-AdminScopeMetadata              | Adds or updates metadata on the given Scope.                                                       |
| Set-AdminServiceMetadata            | Adds or updates metadata on the given Service.                                                     |
| Test-AdminAccess                    | Retrieves the scopes where the specified operation is permitted.                                   |
| Test-AdminDBConnection              | Tests a database connection for the DelegatedAdmin Service.                                        |

# about\_AdminDelegatedAdminSnapin

Jan 04, 2017

## TOPIC

about\_AdminDelegatedAdminSnapin

## SHORT DESCRIPTION

The Delegated Administration Service PowerShell snap-in provides administrative functions for the Delegated Administration Service.

## COMMAND PREFIX

All commands in this snap-in are prefixed with 'Admin'.

## LONG DESCRIPTION

The Delegated Administration Service PowerShell snap-in enables both local and remote administration of the Delegated Administration Service.

The Delegated Administration Service (or DAS for short) stores information about Citrix administrators and the rights they have. Services in the XenDesktop deployment use the DAS to determine whether a particular user has the privilege to perform an operation or not.

The snap-in provides storage and configuration of these entities:

### Administrators

Each administrator object represents an individual person or a group of people identified by their Active Directory account. Administrators can be enabled and disabled.

The effective rights that a user has is the union of any rights that they have by looking at their Active Directory group membership. Disabled administrator entries are ignored for this calculation.

Once a site is setup, there must always be a full administrator and the Delegated Administration snap-in rejects requests to remove or disable the last full administrator.

### Roles

A role represents a job function. That is, anyone with a given role is expected to be able to use or perform the tasks, wizards, and actions associated with that role. Administrators may have multiple roles for a particular site.

Some roles are built-in, and some editions of the product allow custom roles to be created with different combinations of permissions.

## Scopes

Scopes represent a collection of objects, and are used to group objects for administrative purposes in a way that is relevant to the organisation. They can be used to represent both hierarchical and non-hierarchical relationships.

Objects can exist in multiple scopes at once. You may find it easier to think of scopes as labels, or a non-exclusive grouping such as a play-list.

All objects are implicitly in the built-in 'All' scope.

Some objects are not scoped, and access to them is through either the 'All' scope or indirectly through a scoped object. For example sessions are not directly scoped but can be accessed using the scope of the desktop group.

The DAS stores information about scopes, but the mapping between scopes and objects is stored and updated using the PowerShell snap-ins of each corresponding service. For example, Delivery Group scopes are managed using the Broker PowerShell snap-in.

## Rights

Rights determine what an administrator can do and where they can do it. They are expressed as a number of <role, scope> pairs associated with each administrator.

To gain access to any particular object, a person must match an administrator object that has an appropriate right that allows the required operation in a scope that the object is a member of.

## Permissions

Each task, wizard or action in the Citrix Studio or Director consoles represents a unit of functionality that an administrator can perform. Permissions are expressed at a high level and generally correspond directly to the labels in the consoles. For example: "Edit catalog", or "Create delivery group".

## Permission groups:

Permissions are grouped into related functionality when displayed by the console.

## Operations

Operations are the indivisible unit of functionality that each XenDesktop service can perform, and usually correspond to individual cmdlets. Internally, each permission requires a number of operations to be performed, possibly by different services.

# about\_Admin\_Filtering

Jan 04, 2017

## TOPIC

XenDesktop - Advanced Dataset Filtering

## SHORT DESCRIPTION

Describes the common filtering options for XenDesktop cmdlets.

## LONG DESCRIPTION

Some cmdlets operate on large quantities of data and, to reduce the overhead of sending all of that data over the network, many of the Get- cmdlets support server-side filtering of the results.

The conventional way of filtering results in PowerShell is to pipeline them into Where-Object, Select-Object, and Sort-Object, for example:

```
Get-<Noun> | Where { $_.Size = 'Small' } | Sort 'Date' | Select -First 10
```

However, for most XenDesktop cmdlets the data is stored remotely and it would be slow and inefficient to retrieve large amounts of data over the network and then discard most of it. Instead, many of the Get- cmdlets provide filtering parameters that allow results to be processed on the server, returning only the required results.

You can filter results by most object properties using parameters derived from the property name. You can also sort results or limit them to a specified number of records:

```
Get-<Noun> -Size 'Small' -SortBy 'Date' -MaxRecordCount 10
```

You can express more complex filter conditions using a syntax and set of operators very similar to those used by PowerShell expressions.

Those cmdlets that support filtering have the following common parameters:

-MaxRecordCount <int>

Specifies the maximum number of results to return.  
For example, to return only the first nine results use:

```
Get-<Noun> -MaxRecordCount 9
```

If not specified, only the first 250 records are returned, and if more are available, a warning is produced:

WARNING: Only first 250 records returned. Use -MaxRecordCount to retrieve more.

You can suppress this warning by using -WarningAction or by specifying a value for -MaxRecordCount.

To retrieve all records, specify a large number for -MaxRecordCount. As the value is an integer, you can use the following:

```
Get-<Noun> -MaxRecordCount [int]::MaxValue
```

-ReturnTotalRecordCount [<SwitchParameter>]

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. For example:

```
Get-<Noun> -MaxRecordCount 9 -ReturnTotalRecordCount
....
```

```
Get-<Noun> : Returned 9 of 10 items
At line:1 char:18
+ Get-<Noun> <<<< -MaxRecordCount 9 -ReturnTotalRecordCount
+ CategoryInfo          : OperationStopped: (:) [Get-<Noun>], PartialDataException
+ FullyQualifiedErrorId : PartialData,Citrix.<SDKName>.SDK.Get<Noun>
```

The count can be accessed using the TotalAvailableResultCount property:

```
$count = $error[0].TotalAvailableResultCount
```

-Skip <int>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

-SortBy <string>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order, respectively. Ascending order is assumed if no prefix is present.

Sorting occurs before `-MaxRecordCount` and `-Skip` parameters are applied. For example, to sort by Name and then by Count (largest first) use:

```
-SortBy 'Name,-Count'
```

By default, sorting by an enumeration property uses the numeric value of the elements. You can specify a different sort order by qualifying the name with an ordered list of elements or their numeric values, or `<null>` to indicate the placement of null values.

Elements not mentioned are placed at the end in their numeric order. For example, to sort by two different enums and then by the object id:

```
-SortBy 'MyState(StateC,<null>,StateA,StateB),Another(0,3,2,1),Id'
```

```
-Filter <String>
```

This parameter lets you specify advanced filter expressions, and supports combination of conditions with `-and` and `-or`, and grouping with braces. For example:

```
Get-<Noun> -Filter 'Name -like "High*" -or (Priority -eq 1 -and Severity -ge 2)'
```

The syntax is close enough to PowerShell syntax that you can use script blocks in most cases. This can be easier to read as it reduces quoting:

```
Get-<Noun> -Filter { Count -ne $null }
```

The full `-Filter` syntax is provided below.

## EXAMPLES

Filtering by strings performs a case-insensitive wildcard match.

Separate parameters are combined with an implicit `-and` operator.

Normal PowerShell quoting rules apply, so you can use single or double quotes, and omit the quotes altogether for many strings. The order of parameters does not make any difference. The following are equivalent:

```
Get-<Noun> -Company Citrix -Product Xen*
Get-<Noun> -Company "citrix" -Product '[X]EN*'
Get-<Noun> -Product "Xen*" -Company "CITRIX"
Get-<Noun> -Filter { Company -eq 'Citrix' -and Product -like 'Xen*' }
```

See [about\\_Quoting\\_Rules](#) and [about\\_Wildcards](#) for details about PowerShell handling of quotes and wildcards.

To avoid wildcard matching or include quote characters, you can escape the wildcards using the normal PowerShell escape mechanisms (see [about\\_Escape\\_Characters](#)), or switch to a filter expression and the `-eq` operator:

```
Get-<Noun> -Company "Abc[*]"           # Matches Abc*
Get-<Noun> -Company "Abc`*"           # Matches Abc*
Get-<Noun> -Filter { Company -eq "Abc*" } # Matches Abc*
Get-<Noun> -Filter { Company -eq "A`"B`C" } # Matches A"B'C
```

Simple filtering by numbers, booleans, and TimeSpans perform direct equality comparisons, although if the value is nullable you can also search for null values. Here are some examples:

```
Get-<Noun> -Uid 123
Get-<Noun> -Enabled $true
Get-<Noun> -Duration 1:30:40
Get-<Noun> -NullableProperty $null
```

More comparisons are possible using advanced filtering with `-Filter`:

```
Get-<Noun> -Filter 'Capacity -ge 10gb'
Get-<Noun> -Filter 'Age -ge 20 -and Age -lt 40'
Get-<Noun> -Filter 'VolumeLevel -like "[123]"'
Get-<Noun> -Filter 'Enabled -ne $false'
Get-<Noun> -Filter 'NullableProperty -ne $null'
```

You can check boolean values without an explicit comparison operator, and you can also combine them with `-not`:

```
Get-<Noun> -Filter 'Enabled' # Equivalent to 'Enabled -eq $true'
Get-<Noun> -Filter '-not Enabled' # Equivalent to 'Enabled -eq $false'
```

See [about\\_Comparison\\_Operators](#) for an explanation of the operators, but note that only a subset of PowerShell operators are supported (`-eq`, `-ne`, `-gt`, `-ge`, `-lt`, `-le`, `-like`, `-notlike`, `-in`, `-notin`, `-contains`, `-notcontains`).

Enumeration values can either be specified using typed values or the string name of the enumeration value:

```
Get-<Noun> -Shape [Shapes]::Square
Get-<Noun> -Shape Circle
```

With filter expressions, typed values can be specified with simple variables or quoted strings. They also support enumerations with wildcards:



```
$s = [Shapes]::Square
Get-<Noun> -Filter { Shape -eq $s -or Shape -eq "Circle" }
Get-<Noun> -Filter { Shape -like 'C*' }
```

By their nature, floating point values, DateTime values, and TimeSpan values are best suited to relative comparisons rather than just equality. DateTime strings are converted using the locale and time zone of the user device, but you can use ISO8601 format strings (YYYY-MM-DDThh:mm:ss.sTZD) to avoid ambiguity. You can also use standard PowerShell syntax to create these values:

```
Get-<Noun> -Filter { StartTime -ge "2010-08-23T12:30:00.0Z" }
$d = [DateTime]"2010-08-23T12:30:00.0Z"
Get-<Noun> -Filter { StartTime -ge $d }
$d = (Get-Date).AddDays(-1)
Get-<Noun> -Filter { StartTime -ge $d }
```

Relative times are quite common and, when using filter expressions, you can also specify DateTime values using a relative format:

```
Get-<Noun> -Filter { StartTime -ge '-2' } # Two days ago
Get-<Noun> -Filter { StartTime -ge '-1:30' } # Hour and a half ago
Get-<Noun> -Filter { StartTime -ge '-0:0:30' } # 30 seconds ago
```

## ARRAY PROPERTIES

When filtering against list or array properties, simple parameters perform a case-insensitive wildcard match against each of the members. With filter expressions, you can use the -contains and -notcontains operators. Unlike PowerShell, these perform wildcard matching on strings.

Note that for array properties the naming convention is for the returned property to be plural, but the parameter used to search for any match is singular. The following are equivalent (assuming Users is an array property):

```
Get-<Noun> -User Fred*
Get-<Noun> -Filter { User -like "Fred*" }
Get-<Noun> -Filter { Users -contains "Fred*" }
```

You can also use the singular form with -Filter to search using other operators:

```
# Match if any user in the list is called "Frederick"
Get-<Noun> -Filter { User -eq "Frederick" }
# Match if any user in the list has a name alphabetically below 'F'
Get-<Noun> -Filter { User -lt 'F' }
```

## COMPLEX EXPRESSIONS

When matching against multiple values, you can use a sequence of comparisons joined with -or operators, or you can use -in and -notin:

```
Get-<Noun> -Filter { Shape -eq 'Circle' -or Shape -eq 'Square' }
$shapes = 'Circle','Square'
Get-<Noun> -Filter { Shape -in $shapes }
$sides = 1..4
Get-<Noun> -Filter { Sides -notin $sides }
```

Braces can be used to group complex expressions, and override the default left-to-right evaluation of -and and -or. You can also use -not to invert the sense of any sub-expression:

```
Get-<Noun> -Filter { Size -gt 4 -or (Color -eq 'Blue' -and Shape -eq 'Circle') }
Get-<Noun> -Filter { Sides -lt 5 -and -not (Color -eq 'Blue' -and Shape -eq 'Circle') }
```

## PAGING

The simplest way to page through data is to use the -Skip and -MaxRecordCount parameters. So, to read the first three pages of data with 10 records per page, use:

```
Get-<Noun> -Skip 0 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 10 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 20 -MaxRecordCount 10 <other filtering criteria>
```

You must include the same filtering criteria on each call, and ensure that the data is sorted consistently.

The above approach is often acceptable, but as each call performs an independent query, data changes can result in records being skipped or appearing twice. One approach to improve this is to sort by a unique id field and then start the search for the next page at the unique id after the last unique id of the previous page. For example:

```
# Get the first page
Get-<Noun> -MaxRecordCount 10 -SortBy SerialNumber

SerialNumber ...
----- ---
A120004
A120007
... 7 other records ...
A120900

# Get the next page
Get-<Noun> -MaxRecordCount 10 -Filter { FirstName -gt 'A120900' }

SerialNumber ...
```

----- ---  
A120901  
B220000  
...

## FILTER SYNTAX DEFINITION

<Filter> ::= <ScriptBlock> | <ComponentList>

<ScriptBlock> ::= "{" <ComponentList> "}"

<ComponentList> ::= <Component> <AndOrOperator> <ComponentList> |

<Component>

<Component> ::= <NotOperator> <Factor> |

<Factor>

<Factor> ::= "(" <ComponentList> ")" |

<PropertyName> <ComparisonOperator> <Value> |  
<PropertyName>

<AndOrOperator> ::= "-and" | "-or"

<NotOperator> ::= "-not" | "!"

<ComparisonOperator>

::= "-eq" | "-ne" | "-le" | "-ge" | "-lt" | "-gt" |  
"-like" | "-notlike" | "-contains" | "-notcontains" |  
"-in" | "-notin"

<PropertyName> ::= <simple name of property>

<Value> ::= <string literal> | <numeric literal> |

<scalar variable> | <array variable> |  
"\$null" | "\$true" | "\$false"

Numeric literals support decimal and hexadecimal literals, with optional multiplier suffixes (kb, mb, gb, tb, pb).

Dates and times can be specified as string literals. The current culture determines what formats are accepted. To avoid any ambiguity, use strings formatted to the ISO8601 standard. If not specified, the current time zone is used.

Relative date-time string literals are also supported, using a minus sign followed by a TimeSpan. For example, "-1:30" means 1 hour and 30 minutes ago.

# Add-AdminPermission

Jan 04, 2017

Add permissions to the set of permissions of a role.

## Syntax

```
Add-AdminPermission [-InputObject] <Permission[]> -Role <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-AdminPermission [-Permission] <String[]> -Role <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Add extra permissions to the set of permissions that a role maps to.

Any administrator with a right including that role immediately gains the ability to use the operations of the new permissions.

Duplicate permissions do not produce an error, and permissions are skipped if the role already contains the permission (without error).

You cannot modify the permissions of built-in roles.

## Related topics

[Remove-AdminPermission](#)

[Get-AdminPermission](#)

[Get-AdminRole](#)

[Get-AdminPermissionGroup](#)

[Test-AdminAccess](#)

## Parameters

**-InputObject** <Permission[]>

Specifies the permissions to add.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Permission** <String[]>

Specifies the list of permissions to add (by identifier).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Role**<String>

Role name or identifier of the role to update.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Input Type

Citrix.DelegatedAdmin.Sdk.Permission You can pipe a list of permissions to be added into this command.

## Return Values

None

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Add-AdminPermission -Role MyRole -Permission Global_Read,Logging_Read
Add a couple of specific permissions to the 'MyRole' role.
```

----- **EXAMPLE 2** -----

```
C:\PS> $list = Get-AdminRole "Delivery Administrator" | Select -Expand Permissions
C:\PS> Add-AdminPermission -Role MyRole -Permission $list
Add all of the permissions of the Delivery Administrator role to MyRole.
```

# Add-AdminRight

Jan 04, 2017

Grants a given right to the specified administrator.

## Syntax

```
Add-AdminRight -Scope <String> -Role <String> -Administrator <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Add-AdminRight -Role <String> -Administrator <String> [-All] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Add-AdminRight [-InputObject] <Right[]> -Administrator <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Use the Add-AdminRight cmdlet to add rights (role and scope pairs) to an administrator.

For convenience, you can use the -All parameter to specify the 'All' scope.

Use the Get-AdminAdministrator cmdlet to determine what rights an administrator has.

## Related topics

[Get-AdminAdministrator](#)

[Get-AdminEffectiveRight](#)

[Remove-AdminRight](#)

## Parameters

**-InputObject**<Right[]>

Specifies the rights to add from Right objects.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Scope**<String>

Specifies the scope name or scope identifier.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Role**<String>

Specifies the role name or role identifier.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Administrator**<String>

Specifies the name or SID of the administrator.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-All**<SwitchParameter>

Specifies the 'All' scope. This parameter avoids localization issues or having to type the identifier of the 'All' scope.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Input Type

Citrix.DelegatedAdmin.Sdk.Right You can pipe the rights to be added into this command.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Add-AdminRight -Role 'Help Desk Administrator' -Scope London -Administrator DOMAIN\Admin1
Assigns the 'Help Desk Administrator' role and 'London' scope to the 'Admin1' administrator.
```

----- **EXAMPLE 2** -----

```
C:\PS> Add-AdminRight -Role 'Full Administrator' -All -Administrator DOMAIN\Admin1
Assigns the 'Full Administrator' role and 'All' scope to the 'Admin1' administrator.
```

----- **EXAMPLE 3** -----

```
C:\PS> $admin = Get-AdminAdministrator -Name DOMAIN\ExistingAdmin
C:\PS> Add-AdminRight -InputObject $admin.Rights -Administrator DOMAIN\NewAdmin
Copies the administrator rights from 'ExistingAdmin' to 'NewAdmin'.
```

# Get-AdminAdministrator

Jan 04, 2017

Gets administrators configured for this site.

## Syntax

```
Get-AdminAdministrator [[-Name] <String>] [-Sid <String>] [-Enabled <Boolean>] [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves administrators matching the specified criteria. If no parameters are specified this cmdlet enumerates all administrators.

See [about\\_Admin\\_Filtering](#) for information about advanced filtering options.

## Related topics

[New-AdminAdministrator](#)

[Set-AdminAdministrator](#)

[Remove-AdminAdministrator](#)

[Set-AdminAdministratorMetadata](#)

[Remove-AdminAdministratorMetadata](#)

## Parameters

**-Name**<String>

Gets administrators with the specified name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | false                          |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Sid**<String>

Gets administrators with the specified SID (security identifier).

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-Enabled**<Boolean>

Gets administrators with the specified value of Enabled.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Admin\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Admin\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Input Type

None You cannot pipe input into this cmdlet.

Return Values

Citrix.DelegatedAdmin.Sdk.Administrator

Get-AdminAdministrator returns an object for each matching administrator.

Examples

----- **EXAMPLE 1** -----

C:\PS> Get-AdminAdministrator -Name DOMAIN\TestUser  
Finds the administrator object (if one exists) for user "DOMAIN\TestUser".

----- **EXAMPLE 2** -----

C:\PS> Get-AdminAdministrator -Enabled \$false  
Finds all disabled administrator objects.

# Get-AdminDBConnection

Jan 04, 2017

Gets the database string for the specified data store used by the DelegatedAdmin Service.

## Syntax

```
Get-AdminDBConnection [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the database connection string for the specified data store.

If the returned string is blank, no valid connection string has been specified. In this case the service is running, but is idle and awaiting specification of a valid connection string.

## Related topics

[Get-AdminServiceStatus](#)

[Set-AdminDBConnection](#)

[Test-AdminDBConnection](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

system.string

The database connection string configured for the DelegatedAdmin Service.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

NoDBConnections

The database connection string for the DelegatedAdmin Service has not been specified.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

c:\PS>Get-AdminDBConnection

Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True  
Get the database connection string for the DelegatedAdmin Service.



# Get-AdminDBSchema

Jan 04, 2017

Gets a script that creates the DelegatedAdmin Service database schema for the specified data store.

## Syntax

```
Get-AdminDBSchema [-DatabaseName <String>] [-ServiceGroupName <String>] [-ScriptType <ScriptTypes>] [-LocalDatabase] [-Sid <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets SQL scripts that can be used to create a new DelegatedAdmin Service database schema, add a new DelegatedAdmin Service to an existing site, remove a DelegatedAdmin Service from a site, or create a database server logon for a DelegatedAdmin Service. If no Sid parameter is provided, the scripts obtained relate to the currently selected DelegatedAdmin Service instance, otherwise the scripts relate to DelegatedAdmin Service instance running on the machine identified by the Sid provided. When obtaining the Evict script, a Sid parameter must be supplied. The current service instance is that on the local machine, or that explicitly specified by the last usage of the -AdminAddress parameter to a DelegatedAdmin SDK cmdlet. The service instance used to obtain the scripts does not need to be a member of a site or to have had its database connection configured. The database scripts support only Microsoft SQL Server, or SQL Server Express, and require Windows integrated authentication to be used. They can be run using SQL Server's SQLCMD utility, or by copying the script into an SQL Server Management Studio (SSMS) query window and executing the query. If using SSMS, the query must be executed in 'SMDCMD mode'. The ScriptType parameter determines which script is obtained. If ScriptType is not specified, or is FullDatabase, the script contains:

- o Creation of service schema
- o Creation of database server logon
- o Creation of database user
- o Addition of database user to DelegatedAdmin Service roles

If ScriptType is Instance, the returned script contains:

- o Creation of database server logon
- o Creation of database user
- o Addition of database user to DelegatedAdmin Service roles

If ScriptType is Evict, the returned script contains:

- o Removal of DelegatedAdmin Service instance from database
- o Removal of database user

If ScriptType is Login, the returned script contains:

- o Creation of database server logon only

If the service uses two data stores they can exist in the same database. You do not need to configure a database before

using this command.

## Related topics

[Set-AdminDBConnection](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database for which the schema will be generated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ServiceGroupName**<String>

Specifies the name of the service group to be used when creating the database schema. The service group is a collection of all the DelegatedAdmin services that share the same database instance and are considered equivalent; that is, all the services within a service group can be used interchangeably.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Script Type**<ScriptTypes>

Specifies the type of database script returned. Available script types are:

### Database

Returns a full database script that can be used to create a database schema for the DelegatedAdmin Service in a database instance that does not already contain a schema for this service. The DatabaseName and ServiceGroupName parameters must be specified to create a script of this type.

### Instance

Returns a permissions script that can be used to add further DelegatedAdmin services to an existing database instance that already contains the full DelegatedAdmin service schema, associating the services to the Service Group. The Sid parameter can optionally be specified to create a script of this type.

### Login

Returns a database logon script that can be used to add the required logon accounts to an existing database instance that contains the DelegatedAdmin Service schema. This is used primarily when creating a mirrored database environment. The DatabaseName parameter must be specified to create a script of this type.

**Evict**

Returns a script that can be used to remove the specified DelegatedAdmin Service from the database entirely. The DatabaseName and Sid parameters must be specified to create a script of this type.

|                        |          |
|------------------------|----------|
| Required?              | false    |
| Default Value          | Database |
| Accept Pipeline Input? | false    |

**-LocalDatabase**<SwitchParameter>

Specifies whether the database script is to be used in a database instance run on the same controller as other services in the service group. Including this parameter ensures the script creates only the required permissions for local services to access the database schema for DelegatedAdmin services.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Sid**<String>

Specifies the SID of the controller on which the DelegatedAdmin Service instance to remove from the database is running.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

### Systemstring

A string containing the required SQL script for application to a database.

## Notes

The scripts returned support Microsoft SQL Server Express Edition, Microsoft SQL Server Standard Edition, and Microsoft SQL Server Enterprise Edition databases only, and are generated on the assumption that integrated authentication will be used.

If the ScriptType parameter is not included or set to 'FullDatabase', the full database script is returned, which will:

Create the database schema.

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist).

If the ScriptType parameter is set to 'Instance', the script will:

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist) and associate it with a user.

If the ScriptType parameter is set to 'Login', the script will:

Create the logon (providing the schema does not already exist) and associate it with a pre-existing user of the same name.

If the LocalDatabase parameter is included, the NetworkService account will be added to the list of accounts permitted to access the database. This is required only if the database is run on a controller.

If the command fails, the following errors can be returned.

### Error Codes

-----

#### GetSchemasFailed

The database schema could not be found.

#### ActiveDirectoryAccountResolutionFailed

The specified Active Directory account or Group could not be found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AdminDBSchema -DatabaseName MyDB -ServiceGroupName MyServiceGroup > c:\AdminSchema.sql  
Get the full database schema for site data store of the DelegatedAdmin Service and copy it to a file called  
'c:\AdminSchema.sql'.
```

This script can then be used to create the schema in a pre-existing database named 'MyDB' that does not already contain a DelegatedAdmin Service site schema.

----- **EXAMPLE 2** -----

```
c:\PS>Get-AdminDBSchema -DatabaseName MyDB -scriptType Login > c:\DelegatedAdminLogins.sql  
Get the logon scripts for the DelegatedAdmin Service.
```

# Get-AdminDBVersionChangeScript

Jan 04, 2017

Gets a script that updates the DelegatedAdmin Service database schema.

## Syntax

```
Get-AdminDBVersionChangeScript -DatabaseName <String> -TargetVersion <Version> [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Returns a database script that can be used to upgrade or downgrade the site or secondary schema for the DelegatedAdmin Service from the current schema version to a different version.

## Related topics

[Get-AdminInstalledDBVersion](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database instance to which the update applies.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TargetVersion**<Version>

Specifies the version of the database you want to update to.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.Management.Automation.PSObject

A PSObject containing the required SQL script for application to a database.

## Notes

The PSObject returned by this cmdlet contains the following properties:

- Script The raw text of the SQL script to apply the update, or null in the case when no upgrade path to the specified target version exists.
- NeedExclusiveAccess Indicates whether all services in the service group must be shut down during the update or not.
- CanUndo Indicates whether the generated script allows the updated schema to be reverted to the state prior to the update.

Scripts to update the schema version are stored in the database so any service in the service group can obtain these scripts. Extreme caution should be exercised when using update scripts. Citrix recommends backing up the database before attempting to upgrade the schema. Database update scripts may require exclusive use of the schema and so may not be able to execute while any DelegatedAdmin services are running. However, this depends on the specific update being carried out.

After a schema update has been carried out, services that require the previous version of the schema may cease to operate. The ServiceState parameter reported by the Get-AdminServiceStatus command provides information about service compatibility. For example, if the schema has been upgraded to a more recent version that a service cannot use, the service reports "DBNewerVersionThanService".

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoOp

The operation was successful but had no effect.

#### NoDBConnections

The database connection string for the DelegatedAdmin Service has not been specified.

#### DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $update = Get-AdminDBVersionChangeScript -DatabaseName MyDb -TargetVersion 1.0.75.0
```

```
C:\PS> $update.Script > update_75.sql
```

Gets an SQL update script to update the current schema to version 1.0.75.0. The resulting update\_75.sql script is suitable for direct use with the SQL Server SQLCMD utility.



# Get-AdminEffectiveAdministrator

Jan 04, 2017

Retrieve the effective administrator objects for a user.

## Syntax

```
Get-AdminEffectiveAdministrator [-Name] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This command determines what groups the specified user belongs to and retrieves the matching administrator records. It includes the set of rights that would be granted to the user if he or she used the system.

As this command uses Active Directory to determine what groups the user has, the caller must have the ability to read this information from Active Directory.

Only enabled administrator records are returned.

## Related topics

[Get-AdminAdministrator](#)

## Parameters

**-Name**<String>

User name or SID of user to query

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Input Type

string User name or SID of user to query

## Return Values

Citrix.DelegatedAdmin.Sdk.Administrator

Administrator records matching the specified user

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-AdminEffectiveAdministrator MYDOMAIN\testuser
```

Retrieve the administrator records matching user 'testuser'.

# Get-AdminEffectiveRight

Jan 04, 2017

Gets the set of Right objects associated with the current user.

## Syntax

```
Get-AdminEffectiveRight [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Get-AdminEffectiveRight cmdlet returns the effective rights of the current user. This is the union of all rights of the enabled administrators that the current user matches, taking into account Active Directory group membership.

## Related topics

[Get-AdminAdministrator](#)

[Add-AdminRight](#)

[Remove-AdminRight](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

Citrix.DelegatedAdmin.Sdk.Right

The Rights associated with the current user

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-AdminEffectiveRight
```

Return the effective rights for the current user.



# Get-AdminInstalledDBVersion

Jan 04, 2017

Gets a list of all available database schema versions for the DelegatedAdmin Service.

## Syntax

```
Get-AdminInstalledDBVersion [-Upgrade] [-Downgrade] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the current version of the DelegatedAdmin Service database schema, if no flags are set, otherwise returns versions for which upgrade or downgrade scripts are available and have been stored in the database.

## Related topics

## Parameters

**-Upgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be updated should be returned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Downgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be reverted should be returned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

### System.Version

The Get-AdminInstalledDbVersion command returns objects containing the new definition of the DelegatedAdmin Service database schema version.

Major <Integer>

Minor <Integer>

Build <Integer>

Revision <Integer>

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

Both the Upgrade and Downgrade flags were specified.

#### NoOp

The operation was successful but had no effect.

#### NoDBConnections

The database connection string for the DelegatedAdmin Service has not been specified.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AdminInstalledDBVersion
```

Major Minor Build Revision

-----

```
5 6 0 0
```

Get the currently installed version of the DelegatedAdmin Service database schema.

----- **EXAMPLE 2** -----

```
c:\PS>Get-AdminInstalledDBVersion -Upgrade
```

Major Minor Build Revision

-----

```
6 0 0 0
```

Get the versions of the DelegatedAdmin Service database schema for which upgrade scripts are supplied.

# Get-AdminPermission

Jan 04, 2017

Gets permissions configured for the site.

## Syntax

```
Get-AdminPermission [[-Name] <String>] [-Id <String>] [-Description <String>] [-GroupId <String>] [-GroupName <String>] [-Metadata <String>] [-Operation <String>] [-ReadOnly <Boolean>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves permission matching the specified criteria. If no parameters are specified this cmdlet enumerates all permissions.

Permissions are configured using the `Import-AdminRoleConfiguration` command, and are used to represent collections of operations that are needed to perform a particular task. These permissions are presented in Citrix Studio when configuring roles.

Permissions can also have metadata associated with them.

See `about_Admin_Filtering` for information about advanced filtering options.

## Related topics

[Add-AdminPermission](#)

[Remove-AdminPermission](#)

[Get-AdminRole](#)

[Get-AdminPermissionGroup](#)

[Test-AdminAccess](#)

## Parameters

**-Name**<String>

Gets permissions with the specified name (localized)

|                        |                                |
|------------------------|--------------------------------|
| Required?              | false                          |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Id**<String>

Gets permissions with the specified identifier.



|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Description**<String>

Gets permissions with the specified description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-GroupId**<String>

Gets permissions that are a member of the specified permission group (by group id).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-GroupName**<String>

Gets permissions that are a member of the specified permission group (by group name).

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Operation**<String>

Gets permissions that contain a specific operation.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReadOnly**<Boolean>

Gets permissions with the specified value for the ReadOnly flag.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See `about_Admin_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-Sort By**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell-style filter expression. See [about\\_Admin\\_Filtering](#) for details.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.DelegatedAdmin.Sdk.Permission

Get-AdminPermission returns an object for each matching permission.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Get-AdminPermission -Name *Edit*
Finds all permissions with 'Edit' in their names.
```

----- **EXAMPLE 2** -----

```
C:\PS> Get-AdminPermission -Operation "Broker:SetCatalog"
C:\PS> Get-AdminPermission -Filter { Operations -contains "Broker:SetCatalog" -or Operations -contains "Broker:NewCatalog" }
Finds permissions that contain specific operations, with the -Filter form needed to match multiple values.
```

# Get-AdminPermissionGroup

Jan 04, 2017

Gets permission groups configured for the site.

## Syntax

```
Get-AdminPermissionGroup [[-Name] <String>] [-Id <String>] [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves permission groups matching the specified criteria. If no parameters are specified this cmdlet enumerates all permission groups.

Permission groups are configured using the `Import-AdminRoleConfiguration` command, and are primarily used to store the localized name for a group of permissions. Permission groups can also have metadata associated with them.

See `about_Admin_Filtering` for information about advanced filtering options.

## Related topics

[Import-AdminRoleConfiguration](#)

[Get-AdminPermission](#)

## Parameters

**-Name**<String>

Gets permission groups matching the given name. This is the localized name of the group.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | false                          |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Id**<String>

Gets permission groups with the given identifier. This is the non-localized identifier used to associate permissions with permission groups.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Admin\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Admin\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

Citrix.DelegatedAdmin.Sdk.PermissionGroup

Get-AdminPermissionGroup returns an object for each matching permission group.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-AdminPermissionGroup -Name *s
Finds all permission groups that end with the letter 's'.
```

----- **EXAMPLE 2** -----

```
C:\PS> $list = @("Hosts","Other")
C:\PS> Get-AdminPermissionGroup -Filter { Id -in $list } -SortBy Name
Retrieve two specific permission group objects with the matching identifiers.
```



# Get-AdminRevision

Jan 04, 2017

Gets the current revision of the delegated administration configuration data.

## Syntax

```
Get-AdminRevision [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Delegated Administration Service maintains a revision number that is incremented whenever its configuration is changed. This command retrieves the current revision number.

## Related topics

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

System.Int32

The configuration revision number

## Notes

If Int32.MaxValue is reached the value of the revision number will wrap around to Int32.MinValue.

In some cases the value may be incremented by more than one.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-AdminRevision
```

Retrieve the current revision number.



# Get-AdminRole

Jan 04, 2017

Gets roles configured for this site.

## Syntax

```
Get-AdminRole [-Name <String>] [-Id <Guid>] [-BuiltIn <Boolean>] [-Description <String>] [-Metadata <String>] [-Permission <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves roles matching the specified criteria. If no parameters are specified, this cmdlet enumerates all roles.

See [about\\_Admin\\_Filtering](#) for information about advanced filtering options.

## Related topics

[New-AdminRole](#)

[Set-AdminRole](#)

[Rename-AdminRole](#)

[Remove-AdminRole](#)

[Set-AdminRoleMetadata](#)

[Remove-AdminRoleMetadata](#)

## Parameters

**-Name**<String>

Gets roles with the specified name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | false                          |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Id**<Guid>

Gets the role with the specified identifier.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? |       |

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-BuiltIn**<Boolean>

Gets roles with the specified value of the BuiltIn flag.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Gets roles with the specified description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Permission**<String>

Gets roles that contain a specific permission.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Admin\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Admin\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Input Type**

None You cannot pipe input into this cmdlet.

## Return Values

Citrix.DelegatedAdmin.Sdk.Role

Get-AdminRole returns an object for each matching role.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-AdminRole -Id 20852cdf-f527-4953-ba6e-e7545217122d
```

Gets the details of the role with the specific id.

----- **EXAMPLE 2** -----

```
C:\PS> Get-AdminRole -BuiltIn $false
```

List all custom roles.

# Get-AdminRoleConfiguration

Jan 04, 2017

Gets role configurations for this site.

## Syntax

```
Get-AdminRoleConfiguration [-Name] <String> [-Id <Guid>] [-Locale <String>] [-Priority <Int32>] [-Version <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves role configurations matching the specified criteria. If no parameters are specified, this cmdlet enumerates and returns all role configurations.

Role configurations are part of the product configuration and define what permissions, permission groups, and built-in roles the product has. This cmdlet also provides the mapping of permissions to operations.

## Related topics

[Import-AdminRoleConfiguration](#)

## Parameters

**-Name**<String>

Gets role configurations matching the specified name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | false                          |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Id**<Guid>

Gets role configurations with the specified id.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Locale**<String>



Gets role configurations with the specified locale. Role configurations usually have a consistent locale.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Priority**<Int32>

Gets role configurations with the specified priority.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Version**<String>

Gets role configurations with the matching version number.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See `about_Admin_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by `-ReturnTotalRecordCount`.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Admin\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.DelegatedAdmin.Sdk.RoleConfiguration

Get-AdminRoleConfiguration returns an object for each matching role configuration.

**Notes**

This command is supplied for infrastructure purposes only and is not intended for public use.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> Get-AdminRoleConfiguration -Name Director  
Retrieve the role configuration for the Citrix Director component.

# Get-AdminScope

Jan 04, 2017

Gets scopes configured for this site.

## Syntax

```
Get-AdminScope [[-Name] <String>] [-Id <Guid>] [-BuiltIn <Boolean>] [-Description <String>] [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieves scopes matching the specified criteria. If no parameters are specified this cmdlet enumerates all scopes.

There is one special built-in scope, the 'All' scope.

To determine what objects are currently in a scope, use the Get-<Prefix>ScopedObject from each of the relevant PowerShell snap-ins.

See about\_Admin\_Filtering for information about advanced filtering options.

## Related topics

[New-AdminScope](#)

[Set-AdminScope](#)

[Rename-AdminScope](#)

[Remove-AdminScope](#)

[Set-AdminScopeMetadata](#)

[Remove-AdminScopeMetadata](#)

## Parameters

**-Name**<String>

Gets scopes with the specified name.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | false                          |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-Id**<Guid>

Gets the scope with the specified identifier.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-BuiltIn**<Boolean>

Gets scopes with the specified value of the BuiltIn flag.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Description**<String>

Gets scopes with the specified description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Admin\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | 0     |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

**-Filter**<String>

Gets records that match a PowerShell-style filter expression. See [about\\_Admin\\_Filtering](#) for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.DelegatedAdmin.Sdk.Scope

Get-AdminScope returns an object for each matching scope.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Get-AdminScope -Name *Sales*  
List all scopes that contain the word 'Sales'.
```

### ----- EXAMPLE 2 -----

```
C:\PS> Get-AdminScope -Id 21862daf-e529-4553-ba6e-f7543217111e  
Gets the details of the scope with the specific id.
```



# Get-AdminService

Jan 04, 2017

Gets the service record entries for the DelegatedAdmin Service.

## Syntax

```
Get-AdminService [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns instances of the DelegatedAdmin Service that the service publishes. The service records contain account security identifier information that can be used to remove each service from the database.

A database connection for the service is required to use this command.

## Related topics

### Parameters

#### **-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See `about_Admin_Filtering` for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by `-ReturnTotalRecordCount`.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

### **-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Admin\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

Citrix.DelegatedAdmin.Sdk.Service

The Get-AdminServiceInstance command returns an object containing the following properties.

Uid <Integer>

Specifies the unique identifier for the service in the group. The unique identifier is an index number.

ServiceHostId <Guid>

Specifies the unique identifier for the service instance.

DNSName <String>

Specifies the domain name of the host on which the service runs.

MachineName <String>

Specifies the short name of the host on which the service runs.

CurrentState <Citrix.Fma.Sdk.ServiceCore.ServiceState>

Specifies whether the service is running, started but inactive, stopped, or failed.

LastStartTime <DateTime>

Specifies the date and time at which the service was last restarted.

LastActivityTime <DateTime>

Specifies the date and time at which the service was last stopped or restarted.

OSType

Specifies the operating system installed on the host on which the service runs.

OSVersion

Specifies the version of the operating system installed on the host on which the service runs.

ServiceVersion

Specifies the version number of the service instance. The version number is a string that reflects the full build version of the service.

DatabaseUserName <string>

Specifies for the service instance the Active Directory account name with permissions to access the database. This will be either the machine account or, if the database is running on a controller, the NetworkService account.

Sid <string>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

ActiveSiteServices <string[]>

Specifies the names of active site services currently running in the service. Site services are components that perform long-running background processing in some services. This field is empty for services that do not contain site services.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### PartialData

Only a subset of the available data was returned.

#### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

#### CouldNotQueryDatabase

The query required to get the database was not defined.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AdminService
```

```
Uid          : 1
ServiceHostId : aef6f464-f1ee-4042-a523-66982e0cecd0
DNSName      : MyServer.company.com
MachineName  : MYSERVER
CurrentState  : On
LastStartTime : 04/04/2011 15:25:38
LastActivityTime : 04/04/2011 15:33:39
OSType       : Win32NT
OSVersion    : 6.1.7600.0
ServiceVersion : 5.1.0.0
DatabaseUserName : NT AUTHORITY\NETWORK SERVICE
SID          : S-1-5-21-2316621082-1546847349-2782505528-1165
ActiveSiteServices : {MySiteService1, MySiteService2...}
```

Get all the instances of the DelegatedAdmin Service running in the current service group.

# Get-AdminServiceAddedCapability

Jan 04, 2017

Gets any added capabilities for the DelegatedAdmin Service on the controller.

## Syntax

```
Get-AdminServiceAddedCapability [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables updates to the DelegatedAdmin Service on the controller to be detected.

You do not need to configure a database connection before using this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.String

String containing added capabilities.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- **EXAMPLE 1** -----

```
c:\PS>Get-AdminServiceAddedCapability
```

Get the added capabilities of the DelegatedAdmin Service.

# Get-AdminServiceInstance

Jan 04, 2017

Gets the service instance entries for the DelegatedAdmin Service.

## Syntax

```
Get-AdminServiceInstance [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns service interfaces published by the instance of the DelegatedAdmin Service. Each instance of a service publishes multiple interfaces with distinct interface types, and each of these interfaces is represented as a ServiceInstance object. Service instances can be used to register the service with a central configuration service so that other services can use the functionality.

You do not need to configure a database connection to use this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.DelegatedAdmin.Sdk.ServiceInstance

The Get-AdminServiceInstance command returns an object containing the following properties.

ServiceGroupUid <Guid>

Specifies the unique identifier for the service group of which the service is a member.

ServiceGroupName <String>

Specifies the name of the service group of which the service is a member.

ServiceInstanceUID <Guid>

Specifies the unique identifier for registered service instances, which are service instances held by and obtained from a



central configuration service. Unregistered service instances do not have unique identifiers.

ServiceType <String>

Specifies the service instance type. For this service, the service instance type is always Admin.

Address

Specifies the address of the service instance. The address can be used to access the service and, when registered in the central configuration service, can be used by other services to access the service.

Binding

Specifies the binding type that must be used to communicate with the service instance. In this release of XenDesktop, the binding type is always 'wcf\_HTTP\_kerb'. This indicates that the service provides a Windows Communication Foundation endpoint that uses HTTP binding with integrated authentication.

Version

Specifies the version of the service instance. The version number is used to ensure that the correct versions of the services are used for communications.

ServiceAccount <String>

Specifies the Active Directory account name for the machine on which the service instance is running. The account name is used to provide information about the permissions required for interservice communications.

ServiceAccountSid <String>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

InterfaceType <String>

Specifies the interface type. Each service can provide multiple service instances, each for a different purpose, and the interface defines the purpose. Available interfaces are:

SDK - for PowerShell operations

InterService - for operations between different services

Peer - for communications between services of the same type

Metadata <Citrix.DelegatedAdmin.Sdk.Metadata[]>

The collection of metadata associated with registered service instances, which are service instances held by and obtained from a central configuration service. Metadata is not stored for unregistered service instances.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-AdminServiceInstance
```

```
Address      : http://MyServer.com:80/Citrix/DelegatedAdminContract
Binding      : wcf_HTTP_kerb
InterfaceType : SDK
Metadata     :
MetadataMap  :
ServiceAccount : ENG\MyAccount$
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164
ServiceGroupName : MyServiceGroup
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000
ServiceType  : Admin
Version      : 1
```

```
Address      : http://MyServer.com:80/Citrix/DelegatedAdminContract/IServiceApi
Binding      : wcf_HTTP_kerb
InterfaceType : InterService
Metadata     :
MetadataMap  :
```

ServiceAccount : ENG\MyAccount  
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164  
ServiceGroupName : MyServiceGroup  
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d  
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000  
ServiceType : Admin  
Version : 1

Get all instances of the DelegatedAdmin Service running on the specified machine. For remote services, use the AdminAddress parameter to define the service for which the interfaces are required. If the AdminAddress parameter has not been specified for the runspace, service instances running on the local machine are returned.

# Get-AdminServiceStatus

Jan 04, 2017

Gets the current status of the DelegatedAdmin Service on the controller.

## Syntax

```
Get-AdminServiceStatus [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables the status of the DelegatedAdmin Service on the controller to be monitored. If the service has multiple data stores it will return the overall state as an aggregate of all the data store states. For example, if the site data store status is OK and the secondary data store status is DBUnconfigured then it will return DBUnconfigured.

## Related topics

[Set-AdminDBConnection](#)

[Test-AdminDBConnection](#)

[Get-AdminDBConnection](#)

[Get-AdminDBSchema](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Get-AdminServiceStatus command returns an object containing the status of the DelegatedAdmin Service together with extra diagnostics information.

DBUnconfigured

The DelegatedAdmin Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the DelegatedAdmin Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

#### InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the DelegatedAdmin Service schema has not been added to the database.

#### DBNotFound

The specified database could not be located with the configured connection string.

#### DBNewerVersionThanService

The version of the DelegatedAdmin Service currently in use is incompatible with the version of the DelegatedAdmin Service schema on the database. Upgrade the DelegatedAdmin Service to a more recent version.

#### DBOlderVersionThanService

The version of the DelegatedAdmin Service schema on the database is incompatible with the version of the DelegatedAdmin Service currently in use. Upgrade the database schema to a more recent version.

#### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

#### OK

The DelegatedAdmin Service is running and is connected to a database containing a valid schema.

#### Failed

The DelegatedAdmin Service has failed.

#### Unknown

(0) The service status cannot be determined.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AdminServiceStatus
```

DBUnconfigured

Get the current status of the DelegatedAdmin Service.

# Import-AdminRoleConfiguration

Jan 04, 2017

Imports role configuration data into the Delegated Administration Service.

## Syntax

```
Import-AdminRoleConfiguration [-Path] <String> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Import-AdminRoleConfiguration -Content <String> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

This command is intended for use by Citrix Studio to import definitions, roles, permissions, and their mappings to operations.

The supplied configuration requires a digital signature; this is used to validate the integrity of the configuration.

## Related topics

[Get-AdminRoleConfiguration](#)

## Parameters

**-Path**<String>

The path to the file containing the role configuration data.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Content**<String>

The content of the role configuration data.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Force**<SwitchParameter>

Allows older versions of role configuration to replace newer versions.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

#### Input Type

None You cannot pipe input into this cmdlet.

#### Return Values

None

#### Notes

This command is supplied for infrastructure purposes only and is not intended for public use.

#### Examples



----- **EXAMPLE 1** -----

```
C:\PS> Import-AdminRoleConfiguration -Path 'C:\MyAdminConfig.xml'
```

Imports the contents of 'C:\MyAdminConfig.xml' to the Delegated Administration Service.

# New-AdminAdministrator

Jan 04, 2017

Adds a new administrator to the site.

## Syntax

```
New-AdminAdministrator [-Name] <String> [-Enabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
New-AdminAdministrator -Sid <String> [-Enabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

New-AdminAdministrator creates a new administrator object in the site. Once a new administrator has been created you can assign rights (role and scope pairs) to the administrator.

Administrator objects are used to determine what rights, and therefore what permissions a particular Active Directory user has through the various SDKs and consoles of the site.

When the Enabled flag of an administrator is set to false, any rights of the administrator are ignored by the system when performing permission checks.

## Related topics

[Get-AdminAdministrator](#)

[Set-AdminAdministrator](#)

[Remove-AdminAdministrator](#)

[Set-AdminAdministratorMetadata](#)

[Remove-AdminAdministratorMetadata](#)

## Parameters

**-Name**<String>

Specifies the user or group name in Active Directory that this administrator corresponds to.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Sid**<String>

Specifies the SID (security identifier) of the user in Active Directory that this administrator corresponds to.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Enabled**<Boolean>

Specifies whether the new administrator starts off enabled or not.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          | True                  |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Input Type

None You cannot pipe input into this cmdlet.

## Return Values

Citrix.DelegatedAdmin.Sdk.Administrator

The newly created administrator.

## Examples

### ----- **EXAMPLE 1** -----

```
C:\PS> New-AdminAdministrator -Name DOMAIN\TestUser
```

Creates a new administrator object for user "DOMAIN\TestUser". It defaults to enabled.

### ----- **EXAMPLE 2** -----

```
C:\PS> New-AdminAdministrator -Sid S-1-2-34-1234567890-1234567890-1234567890-123
```

Creates a new administrator object for user with SID "S-1-2-34-1234567890-1234567890-1234567890-123". It defaults to enabled.

# New-AdminRole

Jan 04, 2017

Adds a new custom role to the site.

## Syntax

```
New-AdminRole [-Name] <String> [-Description <String>] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

New-AdminRole adds a new custom role object to the site. Once a new role has been created, you can add permissions to the role which define what operations the role conveys.

Roles represent a job function, such as 'help desk administrator', and contain a list of permissions that are required to perform that job function.

To assign a role to an administrator, you combine it with a scope which indicates what objects the role can operate on. This pair (also known as a 'right') can then be assigned to an administrator. See Add-AdminRight for further details.

The identifier of the new role is chosen automatically, and custom roles created with this cmdlet always have their BuiltIn flag set to false.

You cannot modify built-in roles, and only some license editions support custom roles.

## Related topics

[Get-AdminRole](#)

[Set-AdminRole](#)

[Rename-AdminRole](#)

[Remove-AdminRole](#)

[Set-AdminRoleMetadata](#)

[Remove-AdminRoleMetadata](#)

[Add-AdminPermission](#)

[Add-AdminRight](#)

## Parameters

**-Name**<String>

Specifies the name of the role. Each role in a site must have a unique name.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                       |
|------------------------|-----------------------|
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Description**<String>

Specifies the description of the role.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.DelegatedAdmin.Sdk.Role

The newly created role.

## Notes

Roles are created without any permissions. Use the Add-AdminPermission to add permissions.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> New-AdminRole -Name Supervisor -Description "My custom supervisor role"
```

```
C:\PS> $list = Get-AdminRole 'Help Desk Administrator' | Select -Expand Permissions
```

```
C:\PS> Add-AdminPermission -Role Supervisor -Permission $list
```

```
C:\PS> Add-AdminPermission -Role Supervisor -Permission $extras
```

```
C:\PS> Add-AdminRight -Administrator DOMAIN\TestUser -Role Supervisor -All
```

Creates a new role called 'Supervisor', and then copies the permissions from the help desk role and adds some extras. Then gives this role (with the all scope) to user 'TestUser'.

# New-AdminScope

Jan 04, 2017

Adds a new scope to the site.

## Syntax

```
New-AdminScope [-Name] <String> [-Description <String>] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

New-AdminScope adds a new scope object to the site.

A scope represents a collection of objects. Scopes are used to group objects in a way that is relevant to the organization; for example, the set of delivery groups used by the Sales team.

You can create objects in particular scopes by specifying the -Scope parameter of a New- cmdlet for an object that can be scoped. You can then modify the contents of a scope with Add-<Noun>Scope and Remove-<Noun>Scope cmdlets from the corresponding PowerShell snap-ins.

To assign a scope to an administrator, combine it with a role and then assign this pair (also known as a 'right') to an administrator. See Add-AdminRight for further details.

The identifier of the new scope is chosen automatically.

## Related topics

[Get-AdminScope](#)

[Set-AdminScope](#)

[Rename-AdminScope](#)

[Remove-AdminScope](#)

[Set-AdminScopeMetadata](#)

[Remove-AdminScopeMetadata](#)

[Add-AdminRight](#)

## Parameters

**-Name**<String>

Specifies the name of the scope. Each scope in a site must have a unique name.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |



|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-Description**<String>

Specifies the description of the scope.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Input Type**

None You cannot pipe input into this cmdlet.

**Return Values**

Citrix.DelegatedAdmin.Sdk.Scope

The newly created scope.

## Examples

### ----- **EXAMPLE 1** -----

```
C:\PS> New-AdminScope -Name Sales -Description "Sales department scope"
```

```
C:\PS> Add-HypervisorConnectionScope -HypervisorConnectionName XenServer2 -Scope Sales
```

```
C:\PS> Add-AdminRight -Administrator DOMAIN\TestUser -Role Hosting -Scope Sales
```

Creates a new scope called 'Sales', adds a hypervisor connection object to the scope, and then assigns the right to use the hosting role on the Sales scope to the 'TestUser' administrator.

# Remove-AdminAdministrator

Jan 04, 2017

Removes administrators from the site.

## Syntax

```
Remove-AdminAdministrator [-InputObject] <Administrator[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminAdministrator -Sid <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminAdministrator [-Name] <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-AdminAdministrator cmdlet deletes administrators from the site.

## Related topics

[New-AdminAdministrator](#)

[Get-AdminAdministrator](#)

[Set-AdminAdministrator](#)

[Set-AdminAdministratorMetadata](#)

[Remove-AdminAdministratorMetadata](#)

## Parameters

**-InputObject**<Administrator[]>

Specifies the administrators to delete.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String[]>

Specifies the name of the administrator to delete.

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                       |
|------------------------|-----------------------|
| Accept Pipeline Input? | true (ByPropertyName) |
|------------------------|-----------------------|

**-Sid<String[]>**

Specifies the SID of the administrator to delete.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-LoggingId<Guid>**

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Input Type**

Citrix.DelegatedAdmin.Sdk.Administrator You can pipe the administrators to be deleted into this command.

**Return Values**

None

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> Remove-AdminAdministrator DOMAIN\TestUser  
Remove the administrator called "DOMAIN\TestUser".
```

### ----- EXAMPLE 2 -----

```
C:\PS> Get-AdminAdministrator -Enabled $false | Remove-AdminAdministrator  
Remove all disabled administrators.
```

# Remove-AdminAdministratorMetadata

Jan 04, 2017

Removes metadata from the given Administrator.

## Syntax

```
Remove-AdminAdministratorMetadata -AdministratorSid <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminAdministratorMetadata -AdministratorSid <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminAdministratorMetadata [-AdministratorName] <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminAdministratorMetadata [-AdministratorName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminAdministratorMetadata [-InputObject] <Administrator[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminAdministratorMetadata [-InputObject] <Administrator[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Administrator.

## Related topics

[Set-AdminAdministratorMetadata](#)

## Parameters

**-AdministratorName**<String>

Name of the Administrator

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Administrator[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AdministratorSid**<String>

Sid of the Administrator

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-

LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied



You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AdminAdministrator | % { Remove-AdminAdministratorMetadata -Map $_.MetadataMap }
```

Remove all metadata from all Administrator objects.

# Remove-AdminPermission

Jan 04, 2017

Remove permissions from the set of permissions of a role.

## Syntax

```
Remove-AdminPermission [-InputObject] <Permission[]> -Role <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminPermission [-Permission] <String[]> -Role <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminPermission -All -Role <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Remove permissions from the set of permissions that a role maps to.

Any administrator with a right including that role immediately loses the ability to use the operations of the removed permissions.

Duplicate permissions do not produce an error, and permissions that the roles does not already have are skipped (without error).

You cannot modify the permissions of built-in roles.

## Related topics

[Add-AdminPermission](#)

[Get-AdminPermission](#)

[Get-AdminRole](#)

[Get-AdminPermissionGroup](#)

[Test-AdminAccess](#)

## Parameters

**-InputObject** <Permission[]>

Specifies the permissions to remove.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Permission**<String[]>

Specifies the list of permissions to remove (by identifier).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Role**<String>

Role name or identifier of the role to update.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-All**<SwitchParameter>

Remove all permissions.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Input Type**

Citrix.DelegatedAdmin.Sdk.Permission You can pipe a list of permissions to be removed into this command.

**Return Values**

None

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Remove-AdminPermission -Role MyRole -Permission Global_Read,Logging_Read  
Remove a couple of specific permissions from the 'MyRole' role.
```

----- **EXAMPLE 2** -----

```
C:\PS> Remove-AdminPermission -Role MyRole -All  
Remove all permissions from the 'MyRole' role.
```

# Remove-AdminRight

Jan 04, 2017

Removes rights from an administrator.

## Syntax

```
Remove-AdminRight -Scope <String> -Role <String> -Administrator <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminRight -Role <String> -Administrator <String> [-All] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminRight [-InputObject] <Right[]> -Administrator <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

This command removes rights from the specified administrator.

For convenience, you can use the `-All` parameter to specify the 'All' scope.

Use the `Get-AdminAdministrator` cmdlet to determine what rights an administrator has.

## Related topics

[Get-AdminAdministrator](#)

[Get-AdminEffectiveRight](#)

[Add-AdminRight](#)

## Parameters

**-InputObject**<Right[]>

Specifies the rights to remove.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Scope**<String>

Specifies the scope name or scope identifier.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Role**<String>

Specifies the role name or role identifier.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Administrator**<String>

Specifies the name or SID of the administrator.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-All**<SwitchParameter>

Specifies the 'All' scope. This parameter avoids localization issues or having to type the identifier of the 'All' scope.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Input Type

Citrix.DelegatedAdmin.Sdk.Right You can pipe the rights to be removed into this command.

Return Values

None

Examples

----- **EXAMPLE 1** -----

```
C:\PS> RemoveAdminRight -Role 'Help Desk Administrator' -Scope London -Administrator DOMAIN\Admin1
Removes the 'Help Desk Administrator' role and 'London' scope from user 'Admin1'
```

----- **EXAMPLE 2** -----

```
C:\PS> $admin = Get-AdminAdministrator -Name DOMAIN\Admin
C:\PS> Remove-AdminRight -InputObject $admin.Rights -Administrator DOMAIN\Admin
Removes all rights from administrator 'Admin'.
```

# Remove-AdminRole

Jan 04, 2017

Removes a role from the site.

## Syntax

```
Remove-AdminRole [-InputObject] <Role[]> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminRole [-Id] <Guid[]> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminRole [-Name] <String[]> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-AdminRole cmdlet deletes roles from the site.

You cannot remove built-in roles.

An error will be produced if the role being removed is currently assigned to an administrator unless you specify the -Force option. When -Force is specified, any rights that reference the role are also removed.

## Related topics

[New-AdminRole](#)

[Get-AdminRole](#)

[Set-AdminRole](#)

[Rename-AdminRole](#)

[Set-AdminRoleMetadata](#)

[Remove-AdminRoleMetadata](#)

## Parameters

**-InputObject**<Role[]>

Specifies the roles to remove (by role object).

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Id**<Guid[]>

Specifies the roles to remove (by role id).



|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Name**<String[]>

Specifies the roles to remove (by role name).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Force**<SwitchParameter>

Allow removal of roles that are still in use.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name

or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Input Type

Citrix.DelegatedAdmin.Sdk.Role You can pipe the roles to be deleted into this command.

### Return Values

None

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-AdminRole -Name Supervisor  
Remove the Supervisor role.
```

----- **EXAMPLE 2** -----

```
C:\PS> Get-AdminRole -BuiltIn $false | Remove-AdminRole  
Attempt to remove all custom roles. This fails if one of the roles is assigned to an administrator.
```

# Remove-AdminRoleMetadata

Jan 04, 2017

Removes metadata from the given Role.

## Syntax

```
Remove-AdminRoleMetadata [-RoleId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminRoleMetadata [-RoleId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminRoleMetadata [-RoleName] <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminRoleMetadata [-RoleName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminRoleMetadata [-InputObject] <Role[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminRoleMetadata [-InputObject] <Role[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Role.

## Related topics

[Set-AdminRoleMetadata](#)

## Parameters

**-RoleId**<Guid>

Id of the Role

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-RoleName**<String>

Name of the Role

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Role[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-

LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AdminRole | % { Remove-AdminRoleMetadata -Map $_.MetadataMap }
```

Remove all metadata from all Role objects.

# Remove-AdminScope

Jan 04, 2017

Removes a scope from the site.

## Syntax

```
Remove-AdminScope [-InputObject] <Scope[]> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminScope [-Id] <Guid[]> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminScope [-Name] <String[]> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Remove-AdminScope cmdlet deletes scopes from the site.

You cannot remove the built-in 'All' scope.

An error will be produced if the scope being removed is currently assigned to an administrator unless you specify the -Force option. When -Force is specified, any rights that reference the scope are also removed.

## Related topics

[New-AdminScope](#)

[Get-AdminScope](#)

[Set-AdminScope](#)

[Rename-AdminScope](#)

[Set-AdminScopeMetadata](#)

[Remove-AdminScopeMetadata](#)

## Parameters

**-InputObject** <Scope[]>

Specifies the scopes to remove (by scope object).

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Id** <Guid[]>

Specifies the scopes to remove (by scope id).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Name**<String[]>

Specifies the scopes to remove (by scope name).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Force**<SwitchParameter>

Allow removal of scopes that are still in use.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>



Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Input Type

Citrix.DelegatedAdmin.Sdk.Scope You can pipe the scopes to be deleted into this command.

### Return Values

None

### Examples

----- **EXAMPLE 1** -----

```
C:\PS> Remove-AdminScope -Name Sales  
Remove the Sales scope.
```

----- **EXAMPLE 2** -----

```
C:\PS> Get-AdminScope -BuiltIn $false | Remove-AdminScope  
Attempt to remove all scopes (excluding the built-in 'All' scope). This fails if one of the scopes is assigned to an administrator.
```

# Remove-AdminScopeMetadata

Jan 04, 2017

Removes metadata from the given Scope.

## Syntax

```
Remove-AdminScopeMetadata [-ScopeId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminScopeMetadata [-ScopeId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminScopeMetadata [-ScopeName] <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminScopeMetadata [-ScopeName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminScopeMetadata [-InputObject] <Scope[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminScopeMetadata [-InputObject] <Scope[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Scope.

## Related topics

[Set-AdminScopeMetadata](#)

## Parameters

**-ScopeId**<Guid>

Id of the Scope

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-ScopeName**<String>

Name of the Scope

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Scope[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-

LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-AdminScope | % { Remove-AdminScopeMetadata -Map $_.MetadataMap }
```

Remove all metadata from all Scope objects.

# Remove-AdminServiceMetadata

Jan 04, 2017

Removes metadata from the given Service.

## Syntax

```
Remove-AdminServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Remove-AdminServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Remove-AdminServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Remove-AdminServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Service.

## Related topics

[Set-AdminServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Service[]>

Objects to which metadata is to be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples



----- **EXAMPLE 1** -----

```
c:\PS>Get-AdminService | % { Remove-AdminServiceMetadata -Map $_.MetadataMap }  
Remove all metadata from all Service objects.
```

# Rename-AdminRole

Jan 04, 2017

Rename a role

## Syntax

```
Rename-AdminRole [-InputObject] <Role> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Rename-AdminRole [-Id] <Guid> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Rename-AdminRole [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

The Rename-AdminRole cmdlet changes the name of a role.

Role names must be unique, and you cannot modify the name of built-in roles.

## Related topics

[New-AdminRole](#)

[Get-AdminRole](#)

[Set-AdminRole](#)

[Remove-AdminRole](#)

[Set-AdminRoleMetadata](#)

[Remove-AdminRoleMetadata](#)

## Parameters

**-InputObject**<Role>

Specifies the role to rename (by object).

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Id**<Guid>

Specifies the role to rename (by id).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Name**<String>

Specifies the role to rename (by name).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-NewName**<String>

Specifies the new name of the role.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

Returns the affected record. By default, this cmdlet does not generate any output.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create

high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Input Type**

Citrix.DelegatedAdmin.Sdk.Role You can pipe the role to be renamed into this command.

**Return Values**

None or Citrix.DelegatedAdmin.Sdk.Role

When you use the PassThru parameter, Rename-AdminRole generates a Citrix.DelegatedAdmin.Sdk.Role object. Otherwise, this cmdlet does not generate any output.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Rename-AdminRole -Name Supervisor -NewName HelpDeskLead
Renames the 'Supervisor' role to 'HelpDeskLead'.
```

# Rename-AdminScope

Jan 04, 2017

Rename a scope

## Syntax

```
Rename-AdminScope [-InputObject] <Scope> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-AdminScope [-Id] <Guid> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-AdminScope [-Name] <String> [-NewName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Rename-AdminScope cmdlet changes the name of a scope.

Scope names must be unique, and you cannot modify the name of the built-in 'All' scope.

## Related topics

[New-AdminScope](#)

[Get-AdminScope](#)

[Set-AdminScope](#)

[Remove-AdminScope](#)

[Set-AdminScopeMetadata](#)

[Remove-AdminScopeMetadata](#)

## Parameters

**-InputObject**<Scope>

Specifies the scope to rename (by object).

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Id**<Guid>

Specifies the scope to rename (by id).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Name**<String>

Specifies the scope to rename (by name).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-NewName**<String>

Specifies the new name of the scope.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

Returns the affected record. By default, this cmdlet does not generate any output.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create

high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Input Type**

Citrix.DelegatedAdmin.Sdk.Scope You can pipe the scope to be renamed into this command.

**Return Values**

None or Citrix.DelegatedAdmin.Sdk.Scope

When you use the PassThru parameter, Rename-AdminScope generates a Citrix.DelegatedAdmin.Sdk.Scope object. Otherwise, this cmdlet does not generate any output.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS> Rename-AdminScope -Name Sales -NewName SalesDesktops
Renames the 'Sales' scope to 'SalesDesktops'.
```

# Reset-AdminServiceGroupMembership

Jan 04, 2017

Reloads the access permissions and configuration service locations for the DelegatedAdmin Service.

## Syntax

```
Reset-AdminServiceGroupMembership [-ConfigServiceInstance] <ServiceInstance[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables you to reload DelegatedAdmin Service access permissions and configuration service locations. The Reset-AdminServiceGroupMembership command must be run on at least one instance of the service type (Admin) after installation and registration with the configuration service. Without this operation, the DelegatedAdmin services will be unable to communicate with other services in the XenDesktop deployment. When the command is run, the services are updated when additional services are added to the deployment, provided that the configuration service is not stopped. The Reset-AdminServiceGroupMembership command can be run again to refresh this information if automatic updates do not occur when new services are added to the deployment. If more than one configuration service instance is passed to the command, the first instance that meets the expected service type requirements is used.

## Related topics

### Parameters

#### **-ConfigServiceInstance**<ServiceInstance[]>

Specifies the configuration service instance object that represents the service instance for the type 'InterService' that references a configuration service for the deployment.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

#### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | localhost. Once a value is provided by any cmdlet, this value becomes the default. |



|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Input Type

Citrix.DelegatedAdmin.Sdk.ServiceInstance[] Service instances containing a ServiceInstance object that refers to the central configuration service interservice interface can be piped to the Reset-AdminServiceGroupMembership command.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoSuitableServiceInstance

None of the supplied service instance objects were suitable for resetting service group membership.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-AdminServiceGroupMembership
```

Reset the service group membership for a service in a deployment where the configuration service is configured and running on the same machine as the service.

----- **EXAMPLE 2** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config -AdminAddress OtherServer.example.com | Reset-AdminServiceGroupmembership
```

Reset the service group membership for a service in a deployment where the configuration service that is configured and running on a machine

named 'OtherServer.example.com'.

# Set-AdminAdministrator

Jan 04, 2017

Sets the properties of an administrator.

## Syntax

```
Set-AdminAdministrator [-InputObject] <Administrator[]> [-Enabled <Boolean>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminAdministrator -Sid <String[]> [-Enabled <Boolean>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminAdministrator [-Name] <String[]> [-Enabled <Boolean>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-AdminAdministrator cmdlet is used to enable or disable an existing administrator.

You can specify the administrators to modify in a number of ways, by piping in existing objects, by passing existing objects with the InputObject parameter, or by specifying the names or SIDs explicitly.

## Related topics

[New-AdminAdministrator](#)

[Get-AdminAdministrator](#)

[Remove-AdminAdministrator](#)

[Set-AdminAdministratorMetadata](#)

[Remove-AdminAdministratorMetadata](#)

## Parameters

**-InputObject**<Administrator[]>

Specifies the administrators to modify.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String[]>

Specifies the names of the administrators to modify.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Sid**<String[]>

Specifies the SIDs of the administrators to modify.

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Enabled**<Boolean>

Specifies the new value for the Enabled property.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

Returns the affected record. By default, this cmdlet does not generate any output.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Input Type

Citrix.DelegatedAdmin.Sdk.Administrator You can pipe the administrators to be modified into this command.

Return Values

None or Citrix.DelegatedAdmin.Sdk.Administrator

When you use the PassThru parameter, Set-AdminAdministrator generates a Citrix.DelegatedAdmin.Sdk.Administrator object. Otherwise, this cmdlet does not generate any output.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-AdminAdministrator -Enabled $false | Set-AdminAdministrator -Enabled $true
Enable all administrators that are currently disabled.
```

----- **EXAMPLE 2** -----

```
C:\PS> Set-AdminAdministrator -Name DOMAIN\TestUser1,DOMAIN\TestUser2 -Enabled $true
Enable two specific users specified by name (TestUser1 and TestUser2).
```

# Set-AdminAdministratorMetadata

Jan 04, 2017

Adds or updates metadata on the given Administrator.

## Syntax

```
Set-AdminAdministratorMetadata -AdministratorSid <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminAdministratorMetadata -AdministratorSid <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminAdministratorMetadata [-AdministratorName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminAdministratorMetadata [-AdministratorName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminAdministratorMetadata [-InputObject] <Administrator[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminAdministratorMetadata [-InputObject] <Administrator[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given Administrator objects.

## Related topics

[Remove-AdminAdministratorMetadata](#)

## Parameters

**-AdministratorName**<String>

Name of the Administrator

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Administrator[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AdministratorSid**<String>

Sid of the Administrator

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Administrator specified. The property cannot contain any of the following characters \/:#.\*?=<>|[]()"

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

System.Collections.Generic.Dictionary[String,String]

Set-AdminAdministratorMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

**Notes**

If the command fails, the following errors can be returned.

**Error Codes**

-----

**InvalidParameterCombination**

The cmdlet parameters are inconsistent.

**UnknownObject**

One of the specified objects was not found.

**DatabaseError**

An error occurred in the service while attempting a database operation.

**DatabaseNotConfigured**

The operation could not be completed because the database for the service is not configured.

**DataStoreException**

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.



PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-AdminAdministratorMetadata -AdministratorSid S-1-5-21-1505241163-3345470479-1241728991-1000 -Name property -Value value
```

| Key      | Value |
|----------|-------|
| ---      | ----- |
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the Administrator with the identifier 'S-1-5-21-1505241163-3345470479-1241728991-1000'.

# Set-AdminDBConnection

Jan 04, 2017

Configures a database connection for the DelegatedAdmin Service.

## Syntax

```
Set-AdminDBConnection [-DBConnection] <String> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Configures a connection to a database in which the DelegatedAdmin Service can store its state. The service will attempt to connect and start using the database immediately after the connection is configured. The database connection string is updated to the specified value regardless of whether it is valid or not. Specifying an invalid connection string prevents a service from functioning until the error is corrected.

After a connection is configured, you cannot alter it without first clearing it (by setting the connection to \$null).

You do not need to configure a database connection to use this command.

## Related topics

[Get-AdminServiceStatus](#)

[Get-AdminDBConnection](#)

[Test-AdminDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be used by the DelegatedAdmin Service. Passing in \$null will clear any existing database connection configured.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Force**<SwitchParameter>

If present, allows the local administrator to set the connection string to null when there are problems contacting the database or other services.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Set-AdminDBConnection command returns an object containing the status of the DelegatedAdmin Service together with extra diagnostics information.

DBUnconfigured

The DelegatedAdmin Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the DelegatedAdmin Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the DelegatedAdmin Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the DelegatedAdmin Service currently in use is incompatible with the version of the DelegatedAdmin Service schema on the database. Upgrade the DelegatedAdmin Service to a more recent version.

DBOlderVersionThanService

The version of the DelegatedAdmin Service schema on the database is incompatible with the version of the DelegatedAdmin Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The DelegatedAdmin Service is running and is connected to a database containing a valid schema.

Failed

The DelegatedAdmin Service has failed.

Unknown

The status of the DelegatedAdmin Service cannot be determined.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidDBConnectionString

The database connection string has an invalid format.

DatabaseConnectionDetailsAlreadyConfigured

There was already a database connection configured. After a configuration is set, it can only be set to \$null.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-AdminDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Configures a database connection string for the DelegatedAdmin Service.

----- **EXAMPLE 2** -----

```
c:\PS>Set-AdminDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Configures an invalid database connection string for the DelegatedAdmin Service.

# Set-AdminRole

Jan 04, 2017

Set the properties of a role.

## Syntax

```
Set-AdminRole [-InputObject] <Role[]> [-Description <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminRole [-Id] <Guid[]> [-Description <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminRole [-Name] <String[]> [-Description <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-AdminRole command allows the description of custom roles to be updated. You cannot modify built-in roles.

To modify the permissions of a role, use the Add-AdminPermission and Remove-AdminPermission cmdlets.

To update the metadata associated with a role, use the Set-AdminRoleMetadata and Remove-AdminRoleMetadata cmdlets.

## Related topics

[New-AdminRole](#)

[Get-AdminRole](#)

[Remove-AdminRole](#)

[Rename-AdminRole](#)

[Set-AdminRoleMetadata](#)

[Remove-AdminRoleMetadata](#)

[Add-AdminPermission](#)

[Remove-AdminPermission](#)

## Parameters

**-InputObject** <Role[]>

Specifies the roles to update (by object).

|               |      |
|---------------|------|
| Required?     | true |
| Default Value |      |

|                        |                |
|------------------------|----------------|
| Accept Pipeline Input? | true (ByValue) |
|------------------------|----------------|

**-Id**<Guid[]>

Specifies the roles to update (by id).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Name**<String[]>

Specifies the roles to update (by name).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Description**<String>

Supplies the new description value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

Returns the affected record. By default, this cmdlet does not generate any output.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Input Type

Citrix.DelegatedAdmin.Sdk.Role You can pipe the roles to be updated into this command.

### Return Values

None or Citrix.DelegatedAdmin.Sdk.Role

When you use the PassThru parameter, Set-AdminRole generates a Citrix.DelegatedAdmin.Sdk.Role object. Otherwise, this cmdlet does not generate any output.

### Examples

#### ----- **EXAMPLE 1** -----

```
C:\PS> Set-AdminRole -Name Supervisor -Description "Helpdesk supervisor role"
```

Change the description of the 'Supervisor' role.

# Set-AdminRoleMetadata

Jan 04, 2017

Adds or updates metadata on the given Role.

## Syntax

```
Set-AdminRoleMetadata [-RoleId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-AdminRoleMetadata [-RoleId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-AdminRoleMetadata [-RoleName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-AdminRoleMetadata [-RoleName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-AdminRoleMetadata [-InputObject] <Role[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-AdminRoleMetadata [-InputObject] <Role[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given Role objects.

## Related topics

[Remove-AdminRoleMetadata](#)

## Parameters

**-RoleId**<Guid>

Id of the Role

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-RoleName**<String>

Name of the Role



|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Role[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Role specified. The property cannot contain any of the following characters \/:#.\*?=<>|[]()"

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

System.Collections.Generic.Dictionary[String,String]

Set-AdminRoleMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

**Notes**

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-AdminRoleMetadata -RoleId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Key      | Value |
|----------|-------|
| ---      | ----- |
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the Role with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Set-AdminScope

Jan 04, 2017

Set the properties of a scope.

## Syntax

```
Set-AdminScope [-InputObject] <Scope[]> [-Description <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminScope [-Id] <Guid[]> [-Description <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminScope [-Name] <String[]> [-Description <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The Set-AdminScope command allows the description of scopes to be updated. You cannot modify the built-in 'All' scope.

To change the contents of a scope, use the Add-<Noun>Scope and Remove-<Noun>Scope cmdlets from the corresponding PowerShell snap-in.

To update the metadata associated with a scope, use the Set-AdminScopeMetadata and Remove-AdminScopeMetadata cmdlets.

## Related topics

[New-AdminScope](#)

[Get-AdminScope](#)

[Remove-AdminScope](#)

[Rename-AdminScope](#)

[Set-AdminScopeMetadata](#)

[Remove-AdminScopeMetadata](#)

## Parameters

**-InputObject** <Scope[]>

Specifies the scopes to update (by object).

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Id**<Guid[]>

Specifies the scopes to update (by id).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Name**<String[]>

Specifies the scopes to update (by name).

|                        |                       |
|------------------------|-----------------------|
| Required?              | true                  |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Description**<String>

Supplies the new description value.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-PassThru**<SwitchParameter>

Returns the affected record. By default, this cmdlet does not generate any output.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Input Type

Citrix.DelegatedAdmin.Sdk.Scope You can pipe the scopes to be updated into this command.

### Return Values

None or Citrix.DelegatedAdmin.Sdk.Scope

When you use the PassThru parameter, Set-AdminScope generates a Citrix.DelegatedAdmin.Sdk.Scope object. Otherwise, this cmdlet does not generate any output.

### Examples

#### ----- **EXAMPLE 1** -----

```
C:\PS> Set-AdminScope -Name Sales -Description "Sales department desktops"
```

Change the description of the 'Sales' scope.

# Set-AdminScopeMetadata

Jan 04, 2017

Adds or updates metadata on the given Scope.

## Syntax

```
Set-AdminScopeMetadata [-ScopeId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-AdminScopeMetadata [-ScopeId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-AdminScopeMetadata [-ScopeName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-AdminScopeMetadata [-ScopeName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-AdminScopeMetadata [-InputObject] <Scope[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-AdminScopeMetadata [-InputObject] <Scope[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given Scope objects.

## Related topics

[Remove-AdminScopeMetadata](#)

## Parameters

**-ScopeId**<Guid>

Id of the Scope

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-ScopeName**<String>

Name of the Scope

|  |  |
|--|--|
|  |  |
|--|--|



|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Scope[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Scope specified. The property cannot contain any of the following characters \/:#.\*?=<>|[]()"

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

System.Collections.Generic.Dictionary[String,String]

Set-AdminScopeMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

**Notes**

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-AdminScopeMetadata -Scopelid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Key      | Value |
|----------|-------|
| ---      | ----- |
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the Scope with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Set-AdminServiceMetadata

Jan 04, 2017

Adds or updates metadata on the given Service.

## Syntax

```
Set-AdminServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-AdminServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Allows you to store additional custom data against given Service objects.

## Related topics

[Remove-AdminServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Service[]>

Objects to which metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Service specified. The property cannot contain any of the following characters \/:;#.\*?=<>|[]()"

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.Collections.Generic.Dictionary[String,String]

Set-AdminServiceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various

reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-AdminServiceMetadata -ServiceHostId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Key      | Value |
|----------|-------|
| ---      | ----- |
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the Service with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.



# Test-AdminAccess

Jan 04, 2017

Retrieves the scopes where the specified operation is permitted.

## Syntax

```
Test-AdminAccess [-Operation] <String[]> [-Annotate] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This cmdlet evaluates what rights the current user has, and from these determines the scopes where the specified operation is permitted.

Operations are the indivisible unit of functionality that each XenDesktop service can perform, and usually correspond to individual cmdlets.

If you specify the `-Annotate` option or specify multiple operations to check, the resulting object is annotated with the operation the result relates to.

## Related topics

### Parameters

**-Operation**<String[]>

The operation to query.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Annotate**<SwitchParameter>

Annotates each result with the operation it relates to.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name

or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.DelegatedAdmin.Sdk.ScopeReference

The list of permissible scopes for the specified single operation.PSObject

The list of permissible scopes for each operation. This type of object is returned when the -Annotate option or multiple operations are specified.

## Notes

If the specified operation has unrestricted access a single object is returned representing the 'All' scope with a ScopeId of Guid.Empty (00000000-0000-0000-0000-000000000000).

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> Test-AdminAccess -Operation 'Broker:GetCatalog'  
Queries the scopes where 'Broker:GetCatalog' is permitted.
```

----- **EXAMPLE 2** -----

```
C:\PS> Test-AdminAccess -Operation 'Broker:GetCatalog','Broker:GetMachine'  
Queries the scopes where 'Broker:GetCatalog' or 'Broker:GetMachine' are permitted.
```

# Test-AdminDBConnection

Jan 04, 2017

Tests a database connection for the DelegatedAdmin Service.

## Syntax

```
Test-AdminDBConnection [-DBConnection] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Tests a connection to the database in which the DelegatedAdmin Service can store its state. The service will attempt to connect to the database without affecting the current connection to the database.

You do not have to clear the connection to use this command.

## Related topics

[Get-AdminServiceStatus](#)

[Get-AdminDBConnection](#)

[Set-AdminDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be tested by the DelegatedAdmin Service.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

### Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Test-AdminDBConnection command returns an object containing the status of the DelegatedAdmin Service if the connection string of the specified data store were to be set to the string being tested, together with extra diagnostics information for the specified connection string.

### DBRejectedConnection

The database rejected the logon attempt from the DelegatedAdmin Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

### InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the DelegatedAdmin Service schema has not been added to the database.

### DBNotFound

The specified database could not be located with the configured connection string.

### DBNewerVersionThanService

The version of the DelegatedAdmin Service currently in use is incompatible with the version of the DelegatedAdmin Service schema on the database. Upgrade the DelegatedAdmin Service to a more recent version.

### DBOlderVersionThanService

The version of the DelegatedAdmin Service schema on the database is incompatible with the version of the DelegatedAdmin Service currently in use. Upgrade the database schema to a more recent version.

### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

### OK

The Set-AdminDBConnection command would succeed if it were executed with the supplied connection string.

### Failed

The DelegatedAdmin Service has failed.

### Unknown

The status of the DelegatedAdmin Service cannot be determined.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

### InvalidDBConnectionString

The database connection string has an invalid format.

### PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Test-AdminDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Tests a database connection string for the DelegatedAdmin Service.

----- **EXAMPLE 2** -----

```
c:\PS>Test-AdminDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Tests an invalid database connection string for the DelegatedAdmin Service.

# Citrix.EnvTest.Admin.V1

Jan 04, 2017  
Overview

| Name                                 | Description                                                                                                    |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------|
| <a href="#">EnvTestEnvTestSnapin</a> | The Citrix Environment Test Service provides tools to test and inspect the state of a XenDesktop installation. |
| <a href="#">EnvTest Filtering</a>    | Describes the common filtering options for XenDesktop cmdlets.                                                 |

## Cmdlets

| Name                                              | Description                                                                                  |
|---------------------------------------------------|----------------------------------------------------------------------------------------------|
| <a href="#">Get-EnvTestConfiguration</a>          | Gets the Environment Test Service's configuration options                                    |
| <a href="#">Get-EnvTestDBConnection</a>           | Gets the database string for the specified data store used by the EnvTest Service.           |
| <a href="#">Get-EnvTestDBSchema</a>               | Gets a script that creates the EnvTest Service database schema for the specified data store. |
| <a href="#">Get-EnvTestDBVersionChangeScript</a>  | Gets a script that updates the EnvTest Service database schema.                              |
| <a href="#">Get-EnvTestDefinition</a>             | Gets the one or more test definitions                                                        |
| <a href="#">Get-EnvTestInstalledDBVersion</a>     | Gets a list of all available database schema versions for the EnvTest Service.               |
| <a href="#">Get-EnvTestService</a>                | Gets the service record entries for the EnvTest Service.                                     |
| <a href="#">Get-EnvTestServiceAddedCapability</a> | Gets any added capabilities for the EnvTest Service on the controller.                       |
| <a href="#">Get-EnvTestServiceInstance</a>        | Gets the service instance entries for the EnvTest Service.                                   |
| <a href="#">Get-EnvTestServiceStatus</a>          | Gets the current status of the EnvTest Service on the controller.                            |
| <a href="#">Get-EnvTestSuiteDefinition</a>        | Gets one or more test suite definitions.                                                     |
| <a href="#">Get-EnvTestTask</a>                   | Gets one or more EnvTestTask(s)                                                              |

| Name                                 | Description                                                                                  |
|--------------------------------------|----------------------------------------------------------------------------------------------|
| New-EnvTestDiscoveryTargetDefinition | Creates a new EnvTestDiscoveryTargetDefinition object                                        |
| Remove-EnvTestServiceMetadata        | Removes metadata from the given Service.                                                     |
| Remove-EnvTestTask                   | Removes from the database completed tasks for the EnvTest Service.                           |
| Remove-EnvTestTaskMetadata           | Removes metadata from the given Task.                                                        |
| Reset-EnvTestServiceGroupMembership  | Reloads the access permissions and configuration service locations for the EnvTest Service.  |
| Set-EnvTestConfiguration             | Sets the Environment Test Service's configuration options                                    |
| Set-EnvTestDBConnection              | Configures a database connection for the EnvTest Service.                                    |
| Set-EnvTestServiceMetadata           | Adds or updates metadata on the given Service.                                               |
| Set-EnvTestTaskMetadata              | Adds or updates metadata on the given Task.                                                  |
| Start-EnvTestTask                    | Starts a new test task.                                                                      |
| Stop-EnvTestTask                     | Stops a still running task from completing.                                                  |
| Switch-EnvTestTask                   | Sets the current task that will be returned by a call to Get-EnvTestTask with no parameters. |
| Test-EnvTestDBConnection             | Tests a database connection for the EnvTest Service.                                         |

# about\_EnvTestEnvTestSnapin

Jan 04, 2017

## TOPIC

about\_EnvTestEnvTestSnapin

## SHORT DESCRIPTION

The Citrix Environment Test Service provides tools to test and inspect the state of a XenDesktop installation.

## COMMAND PREFIX

All commands in this snap-in have the noun prefixed with 'EnvTest'.

## LONG DESCRIPTION

The Citrix Environment Test Service provides tools to test and inspect the state of a XenDesktop installation at different points during and after configuration and install.



# about\_EnvTest\_Filtering

Jan 04, 2017

## TOPIC

XenDesktop - Advanced Dataset Filtering

## SHORT DESCRIPTION

Describes the common filtering options for XenDesktop cmdlets.

## LONG DESCRIPTION

Some cmdlets operate on large quantities of data and, to reduce the overhead of sending all of that data over the network, many of the Get- cmdlets support server-side filtering of the results.

The conventional way of filtering results in PowerShell is to pipeline them into Where-Object, Select-Object, and Sort-Object, for example:

```
Get-<Noun> | Where { $_.Size = 'Small' } | Sort 'Date' | Select -First 10
```

However, for most XenDesktop cmdlets the data is stored remotely and it would be slow and inefficient to retrieve large amounts of data over the network and then discard most of it. Instead, many of the Get- cmdlets provide filtering parameters that allow results to be processed on the server, returning only the required results.

You can filter results by most object properties using parameters derived from the property name. You can also sort results or limit them to a specified number of records:

```
Get-<Noun> -Size 'Small' -SortBy 'Date' -MaxRecordCount 10
```

You can express more complex filter conditions using a syntax and set of operators very similar to those used by PowerShell expressions.

Those cmdlets that support filtering have the following common parameters:

-MaxRecordCount <int>

Specifies the maximum number of results to return.  
For example, to return only the first nine results use:

```
Get-<Noun> -MaxRecordCount 9
```

If not specified, only the first 250 records are returned, and if more are available, a warning is produced:

WARNING: Only first 250 records returned. Use -MaxRecordCount to retrieve more.

You can suppress this warning by using -WarningAction or by specifying a value for -MaxRecordCount.

To retrieve all records, specify a large number for -MaxRecordCount. As the value is an integer, you can use the following:

```
Get-<Noun> -MaxRecordCount [int]::MaxValue
```

-ReturnTotalRecordCount [<SwitchParameter>]

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. For example:

```
Get-<Noun> -MaxRecordCount 9 -ReturnTotalRecordCount
```

....

```
Get-<Noun> : Returned 9 of 10 items
```

```
At line:1 char:18
```

```
+ Get-<Noun> <<<< -MaxRecordCount 9 -ReturnTotalRecordCount
```

```
+ CategoryInfo          : OperationStopped: (:) [Get-<Noun>], PartialDataException
```

```
+ FullyQualifiedErrorId : PartialData,Citrix.<SDKName>.SDK.Get<Noun>
```

The count can be accessed using the TotalAvailableResultCount property:

```
$count = $error[0].TotalAvailableResultCount
```

-Skip <int>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

-SortBy <string>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order, respectively. Ascending order is assumed if no prefix is present.

Sorting occurs before `-MaxRecordCount` and `-Skip` parameters are applied. For example, to sort by Name and then by Count (largest first) use:

```
-SortBy 'Name,-Count'
```

By default, sorting by an enumeration property uses the numeric value of the elements. You can specify a different sort order by qualifying the name with an ordered list of elements or their numeric values, or `<null>` to indicate the placement of null values.

Elements not mentioned are placed at the end in their numeric order. For example, to sort by two different enums and then by the object id:

```
-SortBy 'MyState(StateC,<null>,StateA,StateB),Another(0,3,2,1),Id'
```

```
-Filter <String>
```

This parameter lets you specify advanced filter expressions, and supports combination of conditions with `-and` and `-or`, and grouping with braces. For example:

```
Get-<Noun> -Filter 'Name -like "High*" -or (Priority -eq 1 -and Severity -ge 2)'
```

The syntax is close enough to PowerShell syntax that you can use script blocks in most cases. This can be easier to read as it reduces quoting:

```
Get-<Noun> -Filter { Count -ne $null }
```

The full `-Filter` syntax is provided below.

## EXAMPLES

Filtering by strings performs a case-insensitive wildcard match.

Separate parameters are combined with an implicit `-and` operator.

Normal PowerShell quoting rules apply, so you can use single or double quotes, and omit the quotes altogether for many strings. The order of parameters does not make any difference. The following are equivalent:

```
Get-<Noun> -Company Citrix -Product Xen*
Get-<Noun> -Company "citrix" -Product '[X]EN*'
Get-<Noun> -Product "Xen*" -Company "CITRIX"
Get-<Noun> -Filter { Company -eq 'Citrix' -and Product -like 'Xen*' }
```

See [about\\_Quoting\\_Rules](#) and [about\\_Wildcards](#) for details about PowerShell handling of quotes and wildcards.

To avoid wildcard matching or include quote characters, you can escape the wildcards using the normal PowerShell escape mechanisms (see [about\\_Escape\\_Characters](#)), or switch to a filter expression and the `-eq` operator:

```
Get-<Noun> -Company "Abc[*]"           # Matches Abc*
Get-<Noun> -Company "Abc`*"           # Matches Abc*
Get-<Noun> -Filter { Company -eq "Abc*" } # Matches Abc*
Get-<Noun> -Filter { Company -eq "A`"B`C" } # Matches A"B'C
```

Simple filtering by numbers, booleans, and TimeSpans perform direct equality comparisons, although if the value is nullable you can also search for null values. Here are some examples:

```
Get-<Noun> -Uid 123
Get-<Noun> -Enabled $true
Get-<Noun> -Duration 1:30:40
Get-<Noun> -NullableProperty $null
```

More comparisons are possible using advanced filtering with `-Filter`:

```
Get-<Noun> -Filter 'Capacity -ge 10gb'
Get-<Noun> -Filter 'Age -ge 20 -and Age -lt 40'
Get-<Noun> -Filter 'VolumeLevel -like "[123]"'
Get-<Noun> -Filter 'Enabled -ne $false'
Get-<Noun> -Filter 'NullableProperty -ne $null'
```

You can check boolean values without an explicit comparison operator, and you can also combine them with `-not`:

```
Get-<Noun> -Filter 'Enabled' # Equivalent to 'Enabled -eq $true'
Get-<Noun> -Filter '-not Enabled' # Equivalent to 'Enabled -eq $false'
```

See [about\\_Comparison\\_Operators](#) for an explanation of the operators, but note that only a subset of PowerShell operators are supported (`-eq`, `-ne`, `-gt`, `-ge`, `-lt`, `-le`, `-like`, `-notlike`, `-in`, `-notin`, `-contains`, `-notcontains`).

Enumeration values can either be specified using typed values or the string name of the enumeration value:

```
Get-<Noun> -Shape [Shapes]::Square
Get-<Noun> -Shape Circle
```

With filter expressions, typed values can be specified with simple variables or quoted strings. They also support enumerations with wildcards:

```
$s = [Shapes]::Square
Get-<Noun> -Filter { Shape -eq $s -or Shape -eq "Circle" }
Get-<Noun> -Filter { Shape -like 'C*' }
```

By their nature, floating point values, DateTime values, and TimeSpan values are best suited to relative comparisons rather than just equality. DateTime strings are converted using the locale and time zone of the user device, but you can use ISO8601 format strings (YYYY-MM-DDThh:mm:ss.sTZD) to avoid ambiguity. You can also use standard PowerShell syntax to create these values:

```
Get-<Noun> -Filter { StartTime -ge "2010-08-23T12:30:00.0Z" }
$d = [DateTime]"2010-08-23T12:30:00.0Z"
Get-<Noun> -Filter { StartTime -ge $d }
$d = (Get-Date).AddDays(-1)
Get-<Noun> -Filter { StartTime -ge $d }
```

Relative times are quite common and, when using filter expressions, you can also specify DateTime values using a relative format:

```
Get-<Noun> -Filter { StartTime -ge '-2' } # Two days ago
Get-<Noun> -Filter { StartTime -ge '-1:30' } # Hour and a half ago
Get-<Noun> -Filter { StartTime -ge '-0:0:30' } # 30 seconds ago
```

## ARRAY PROPERTIES

When filtering against list or array properties, simple parameters perform a case-insensitive wildcard match against each of the members. With filter expressions, you can use the -contains and -notcontains operators. Unlike PowerShell, these perform wildcard matching on strings.

Note that for array properties the naming convention is for the returned property to be plural, but the parameter used to search for any match is singular. The following are equivalent (assuming Users is an array property):

```
Get-<Noun> -User Fred*
Get-<Noun> -Filter { User -like "Fred*" }
Get-<Noun> -Filter { Users -contains "Fred*" }
```

You can also use the singular form with -Filter to search using other operators:

```
# Match if any user in the list is called "Frederick"
Get-<Noun> -Filter { User -eq "Frederick" }
# Match if any user in the list has a name alphabetically below 'F'
Get-<Noun> -Filter { User -lt 'F' }
```

## COMPLEX EXPRESSIONS

When matching against multiple values, you can use a sequence of comparisons joined with -or operators, or you can use -in and -notin:

```
Get-<Noun> -Filter { Shape -eq 'Circle' -or Shape -eq 'Square' }
$shapes = 'Circle','Square'
Get-<Noun> -Filter { Shape -in $shapes }
$sides = 1..4
Get-<Noun> -Filter { Sides -notin $sides }
```

Braces can be used to group complex expressions, and override the default left-to-right evaluation of -and and -or. You can also use -not to invert the sense of any sub-expression:

```
Get-<Noun> -Filter { Size -gt 4 -or (Color -eq 'Blue' -and Shape -eq 'Circle') }
Get-<Noun> -Filter { Sides -lt 5 -and -not (Color -eq 'Blue' -and Shape -eq 'Circle') }
```

## PAGING

The simplest way to page through data is to use the -Skip and -MaxRecordCount parameters. So, to read the first three pages of data with 10 records per page, use:

```
Get-<Noun> -Skip 0 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 10 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 20 -MaxRecordCount 10 <other filtering criteria>
```

You must include the same filtering criteria on each call, and ensure that the data is sorted consistently.

The above approach is often acceptable, but as each call performs an independent query, data changes can result in records being skipped or appearing twice. One approach to improve this is to sort by a unique id field and then start the search for the next page at the unique id after the last unique id of the previous page. For example:

```
# Get the first page
Get-<Noun> -MaxRecordCount 10 -SortBy SerialNumber

SerialNumber ...
----- ---
A120004
A120007
... 7 other records ...
A120900

# Get the next page
Get-<Noun> -MaxRecordCount 10 -Filter { FirstName -gt 'A120900' }

SerialNumber ...
```

----- ---  
A120901  
B220000  
...

## FILTER SYNTAX DEFINITION

<Filter> ::= <ScriptBlock> | <ComponentList>

<ScriptBlock> ::= "{" <ComponentList> "}"

<ComponentList> ::= <Component> <AndOrOperator> <ComponentList> |

<Component>

<Component> ::= <NotOperator> <Factor> |

<Factor>

<Factor> ::= "(" <ComponentList> ")" |

<PropertyName> <ComparisonOperator> <Value> |  
<PropertyName>

<AndOrOperator> ::= "-and" | "-or"

<NotOperator> ::= "-not" | "!"

<ComparisonOperator>

::= "-eq" | "-ne" | "-le" | "-ge" | "-lt" | "-gt" |  
"-like" | "-notlike" | "-contains" | "-notcontains" |  
"-in" | "-notin"

<PropertyName> ::= <simple name of property>

<Value> ::= <string literal> | <numeric literal> |

<scalar variable> | <array variable> |  
"\$null" | "\$true" | "\$false"

Numeric literals support decimal and hexadecimal literals, with optional multiplier suffixes (kb, mb, gb, tb, pb).

Dates and times can be specified as string literals. The current culture determines what formats are accepted. To avoid any ambiguity, use strings formatted to the ISO8601 standard. If not specified, the current time zone is used.

Relative date-time string literals are also supported, using a minus sign followed by a TimeSpan. For example, "-1:30" means 1 hour and 30 minutes ago.



# Get-EnvTestConfiguration

Jan 04, 2017

Gets the Environment Test Service's configuration options

## Syntax

```
Get-EnvTestConfiguration [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets the Environment Test Service's configuration options and returns them as key/value pairs.

## Related topics

### Parameters

#### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Dictionary<string, object>

All configuration settings

## Examples

----- **EXAMPLE 1** -----

Get-EnvTestConfiguration  
Gets all configuration options

# Get-EnvTestDBConnection

Jan 04, 2017

Gets the database string for the specified data store used by the EnvTest Service.

## Syntax

```
Get-EnvTestDBConnection [-AdminAddress <String>][<CommonParameters>]
```

## Detailed Description

Returns the database connection string for the specified data store.

If the returned string is blank, no valid connection string has been specified. In this case the service is running, but is idle and awaiting specification of a valid connection string.

## Related topics

[Get-EnvTestServiceStatus](#)

[Set-EnvTestDBConnection](#)

[Test-EnvTestDBConnection](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

system.string

The database connection string configured for the EnvTest Service.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

NoDBConnections

The database connection string for the EnvTest Service has not been specified.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-EnvTestDBConnection
```

Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True  
Get the database connection string for the EnvTest Service.

# Get-EnvTestDBSchema

Jan 04, 2017

Gets a script that creates the EnvTest Service database schema for the specified data store.

## Syntax

```
Get-EnvTestDBSchema [-DatabaseName <String>] [-ServiceGroupName <String>] [-ScriptType <ScriptTypes>] [-LocalDatabase] [-Sid <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets SQL scripts that can be used to create a new EnvTest Service database schema, add a new EnvTest Service to an existing site, remove a EnvTest Service from a site, or create a database server logon for a EnvTest Service. If no Sid parameter is provided, the scripts obtained relate to the currently selected EnvTest Service instance, otherwise the scripts relate to EnvTest Service instance running on the machine identified by the Sid provided. When obtaining the Evict script, a Sid parameter must be supplied. The current service instance is that on the local machine, or that explicitly specified by the last usage of the -AdminAddress parameter to a EnvTest SDK cmdlet. The service instance used to obtain the scripts does not need to be a member of a site or to have had its database connection configured. The database scripts support only Microsoft SQL Server, or SQL Server Express, and require Windows integrated authentication to be used. They can be run using SQL Server's SQLCMD utility, or by copying the script into an SQL Server Management Studio (SSMS) query window and executing the query. If using SSMS, the query must be executed in 'SMDCMD mode'. The ScriptType parameter determines which script is obtained. If ScriptType is not specified, or is FullDatabase, the script contains:

- o Creation of service schema
- o Creation of database server logon
- o Creation of database user
- o Addition of database user to EnvTest Service roles

If ScriptType is Instance, the returned script contains:

- o Creation of database server logon
- o Creation of database user
- o Addition of database user to EnvTest Service roles

If ScriptType is Evict, the returned script contains:

- o Removal of EnvTest Service instance from database
- o Removal of database user

If ScriptType is Login, the returned script contains:

- o Creation of database server logon only

If the service uses two data stores they can exist in the same database. You do not need to configure a database before using this command.

## Related topics

[Set-EnvTestDBConnection](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database for which the schema will be generated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ServiceGroupName**<String>

Specifies the name of the service group to be used when creating the database schema. The service group is a collection of all the EnvTest services that share the same database instance and are considered equivalent; that is, all the services within a service group can be used interchangeably.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Script Type**<ScriptTypes>

Specifies the type of database script returned. Available script types are:

### Database

Returns a full database script that can be used to create a database schema for the EnvTest Service in a database instance that does not already contain a schema for this service. The DatabaseName and ServiceGroupName parameters must be specified to create a script of this type.

### Instance

Returns a permissions script that can be used to add further EnvTest services to an existing database instance that already contains the full EnvTest service schema, associating the services to the Service Group. The Sid parameter can optionally be specified to create a script of this type.

### Login

Returns a database logon script that can be used to add the required logon accounts to an existing database instance that contains the EnvTest Service schema. This is used primarily when creating a mirrored database environment. The

DatabaseName parameter must be specified to create a script of this type.

#### Evict

Returns a script that can be used to remove the specified EnvTest Service from the database entirely. The DatabaseName and Sid parameters must be specified to create a script of this type.

|                        |          |
|------------------------|----------|
| Required?              | false    |
| Default Value          | Database |
| Accept Pipeline Input? | false    |

#### -LocalDatabase<SwitchParameter>

Specifies whether the database script is to be used in a database instance run on the same controller as other services in the service group. Including this parameter ensures the script creates only the required permissions for local services to access the database schema for EnvTest services.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### -Sid<String>

Specifies the SID of the controller on which the EnvTest Service instance to remove from the database is running.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

#### -AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

### Systemstring

A string containing the required SQL script for application to a database.

## Notes

The scripts returned support Microsoft SQL Server Express Edition, Microsoft SQL Server Standard Edition, and Microsoft SQL Server Enterprise Edition databases only, and are generated on the assumption that integrated authentication will be used.

If the ScriptType parameter is not included or set to 'FullDatabase', the full database script is returned, which will:

Create the database schema.

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist).

If the ScriptType parameter is set to 'Instance', the script will:

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist) and associate it with a user.

If the ScriptType parameter is set to 'Login', the script will:

Create the logon (providing the schema does not already exist) and associate it with a pre-existing user of the same name.

If the LocalDatabase parameter is included, the NetworkService account will be added to the list of accounts permitted to access the database. This is required only if the database is run on a controller.

If the command fails, the following errors can be returned.

## Error Codes

-----

### GetSchemasFailed

The database schema could not be found.

### ActiveDirectoryAccountResolutionFailed

The specified Active Directory account or Group could not be found.

### DatabaseError

An error occurred in the service while attempting a database operation.



## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-EnvTestDBSchema -DatabaseName MyDB -ServiceGroupName MyServiceGroup > c:\EnvTestSchema.sql  
Get the full database schema for site data store of the EnvTest Service and copy it to a file called 'c:\EnvTestSchema.sql'.
```

This script can then be used to create the schema in a pre-existing database named 'MyDB' that does not already contain a EnvTest Service site schema.

### ----- EXAMPLE 2 -----

```
c:\PS>Get-EnvTestDBSchema -DatabaseName MyDB -scriptType Login > c:\EnvTestLogins.sql  
Get the logon scripts for the EnvTest Service.
```

# Get-EnvTestDBVersionChangeScript

Jan 04, 2017

Gets a script that updates the EnvTest Service database schema.

## Syntax

```
Get-EnvTestDBVersionChangeScript -DatabaseName <String> -TargetVersion <Version> [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Returns a database script that can be used to upgrade or downgrade the site or secondary schema for the EnvTest Service from the current schema version to a different version.

## Related topics

[Get-EnvTestInstalledDBVersion](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database instance to which the update applies.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TargetVersion**<Version>

Specifies the version of the database you want to update to.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.Management.Automation.PSObject

A PSObject containing the required SQL script for application to a database.

## Notes

The PSObject returned by this cmdlet contains the following properties:

- Script The raw text of the SQL script to apply the update, or null in the case when no upgrade path to the specified target version exists.
- NeedExclusiveAccess Indicates whether all services in the service group must be shut down during the update or not.
- CanUndo Indicates whether the generated script allows the updated schema to be reverted to the state prior to the update.

Scripts to update the schema version are stored in the database so any service in the service group can obtain these scripts. Extreme caution should be exercised when using update scripts. Citrix recommends backing up the database before attempting to upgrade the schema. Database update scripts may require exclusive use of the schema and so may not be able to execute while any EnvTest services are running. However, this depends on the specific update being carried out.

After a schema update has been carried out, services that require the previous version of the schema may cease to operate. The ServiceState parameter reported by the Get-EnvTestServiceStatus command provides information about service compatibility. For example, if the schema has been upgraded to a more recent version that a service cannot use, the service reports "DBNewerVersionThanService".

If the command fails, the following errors can be returned.

## Error Codes

-----

### NoOp

The operation was successful but had no effect.

### NoDBConnections

The database connection string for the EnvTest Service has not been specified.

### DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $update = Get-EnvTestDBVersionChangeScript -DatabaseName MyDb -TargetVersion 1.0.75.0
```

```
C:\PS> $update.Script > update_75.sql
```

Gets an SQL update script to update the current schema to version 1.0.75.0. The resulting update\_75.sql script is suitable for direct use with the SQL Server SQLCMD utility.

# Get-EnvTestDefinition

Jan 04, 2017

Gets the one or more test definitions

## Syntax

```
Get-EnvTestDefinition [-TestId <String[]>] [-CultureName <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns a list of test definitions that are available from currently running components.

## Related topics

[Get-EnvTestSuiteDefinition](#)

[Get-EnvTestTask](#)

[Start-EnvTestTask](#)

[Switch-EnvTestTask](#)

[Stop-EnvTestTask](#)

[Remove-EnvTestTask](#)

[Add-EnvTestTaskMetadata](#)

[Remove-EnvTestTaskMetadata](#)

## Parameters

**-TestId**<String[]>

The id of one or more tests.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-CultureName**<String>

The culture name in which to produce results. The culture name is in standard language/region-code format; for example "en-US".

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Input Type

System.String A test id. System.String[] An array of test ids.

Return Values

Citrix.EnvTest.Sdk.EnvTestDefinition

One or more test definitions.

Examples

----- **EXAMPLE 1** -----

```
$allTestDefinitions = Get-EnvTestDefinition
```

Retrieve all tests.

----- **EXAMPLE 2** -----

```
$allTestDefinitionsTranslatedIntoSpanish = Get-EnvTestDefinition -CultureName es-ES
```

Retrieve all tests with localized properties returned in Spanish.

----- **EXAMPLE 3** -----

```
$monitorConfigServiceRegistrationDefinition = Get-EnvTestDefinition -TestId Monitor_RegisteredWithConfigurationService
```

Retrieve the definition of the 'Monitor\_RegisteredWithConfigurationService' test.

# Get-EnvTestInstalledDBVersion

Jan 04, 2017

Gets a list of all available database schema versions for the EnvTest Service.

## Syntax

```
Get-EnvTestInstalledDBVersion [-Upgrade] [-Downgrade] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the current version of the EnvTest Service database schema, if no flags are set, otherwise returns versions for which upgrade or downgrade scripts are available and have been stored in the database.

## Related topics

## Parameters

**-Upgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be updated should be returned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Downgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be reverted should be returned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

### System.Version

The Get-EnvTestInstalledDbVersion command returns objects containing the new definition of the EnvTest Service database schema version.

Major <Integer>

Minor <Integer>

Build <Integer>

Revision <Integer>

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

Both the Upgrade and Downgrade flags were specified.

#### NoOp

The operation was successful but had no effect.

#### NoDBConnections

The database connection string for the EnvTest Service has not been specified.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.



AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-EnvTestInstalledDBVersion
```

Major Minor Build Revision

-----

5 6 0 0

Get the currently installed version of the EnvTest Service database schema.

----- **EXAMPLE 2** -----

```
c:\PS>Get-EnvTestInstalledDBVersion -Upgrade
```

Major Minor Build Revision

-----

6 0 0 0

Get the versions of the EnvTest Service database schema for which upgrade scripts are supplied.

# Get-EnvTestService

Jan 04, 2017

Gets the service record entries for the EnvTest Service.

## Syntax

```
Get-EnvTestService [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns instances of the EnvTest Service that the service publishes. The service records contain account security identifier information that can be used to remove each service from the database.

A database connection for the service is required to use this command.

## Related topics

### Parameters

#### **-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_EnvTest\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

### **-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_EnvTest\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

Citrix.EnvTest.Sdk.Service

The Get-EnvTestServiceInstance command returns an object containing the following properties.

Uid <Integer>

Specifies the unique identifier for the service in the group. The unique identifier is an index number.

ServiceHostId <Guid>

Specifies the unique identifier for the service instance.

DNSName <String>

Specifies the domain name of the host on which the service runs.

MachineName <String>

Specifies the short name of the host on which the service runs.

CurrentState <Citrix.Fma.Sdk.ServiceCore.ServiceState>

Specifies whether the service is running, started but inactive, stopped, or failed.

LastStartTime <DateTime>

Specifies the date and time at which the service was last restarted.

LastActivityTime <DateTime>

Specifies the date and time at which the service was last stopped or restarted.

OSType

Specifies the operating system installed on the host on which the service runs.

OSVersion

Specifies the version of the operating system installed on the host on which the service runs.

ServiceVersion

Specifies the version number of the service instance. The version number is a string that reflects the full build version of the service.

DatabaseUserName <string>

Specifies for the service instance the Active Directory account name with permissions to access the database. This will be either the machine account or, if the database is running on a controller, the NetworkService account.

Sid <string>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

ActiveSiteServices <string[]>

Specifies the names of active site services currently running in the service. Site services are components that perform long-running background processing in some services. This field is empty for services that do not contain site services.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### PartialData

Only a subset of the available data was returned.

#### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

#### CouldNotQueryDatabase

The query required to get the database was not defined.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-EnvTestService
```

```
Uid          : 1
ServiceHostId : aef6f464-f1ee-4042-a523-66982e0cecd0
DNSName      : MyServer.company.com
MachineName  : MYSERVER
CurrentState  : On
LastStartTime : 04/04/2011 15:25:38
LastActivityTime : 04/04/2011 15:33:39
OSType       : Win32NT
OSVersion    : 6.1.7600.0
ServiceVersion : 5.1.0.0
DatabaseUserName : NT AUTHORITY\NETWORK SERVICE
SID          : S-1-5-21-2316621082-1546847349-2782505528-1165
ActiveSiteServices : {MySiteService1, MySiteService2...}
Get all the instances of the EnvTest Service running in the current service group.
```

# Get-EnvTestServiceAddedCapability

Jan 04, 2017

Gets any added capabilities for the EnvTest Service on the controller.

## Syntax

```
Get-EnvTestServiceAddedCapability [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables updates to the EnvTest Service on the controller to be detected.

You do not need to configure a database connection before using this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.String

String containing added capabilities.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- **EXAMPLE 1** -----

```
c:\PS>Get-EnvTestServiceAddedCapability
```

Get the added capabilities of the EnvTest Service.



# Get-EnvTestServiceInstance

Jan 04, 2017

Gets the service instance entries for the EnvTest Service.

## Syntax

```
Get-EnvTestServiceInstance [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns service interfaces published by the instance of the EnvTest Service. Each instance of a service publishes multiple interfaces with distinct interface types, and each of these interfaces is represented as a ServiceInstance object. Service instances can be used to register the service with a central configuration service so that other services can use the functionality.

You do not need to configure a database connection to use this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.EnvTest.Sdk.ServiceInstance

The Get-EnvTestServiceInstance command returns an object containing the following properties.

ServiceGroupUid <Guid>

Specifies the unique identifier for the service group of which the service is a member.

ServiceGroupName <String>

Specifies the name of the service group of which the service is a member.

ServiceInstanceUID <Guid>

Specifies the unique identifier for registered service instances, which are service instances held by and obtained from a

central configuration service. Unregistered service instances do not have unique identifiers.

ServiceType <String>

Specifies the service instance type. For this service, the service instance type is always EnvTest.

Address

Specifies the address of the service instance. The address can be used to access the service and, when registered in the central configuration service, can be used by other services to access the service.

Binding

Specifies the binding type that must be used to communicate with the service instance. In this release of XenDesktop, the binding type is always 'wcf\_HTTP\_kerb'. This indicates that the service provides a Windows Communication Foundation endpoint that uses HTTP binding with integrated authentication.

Version

Specifies the version of the service instance. The version number is used to ensure that the correct versions of the services are used for communications.

ServiceAccount <String>

Specifies the Active Directory account name for the machine on which the service instance is running. The account name is used to provide information about the permissions required for interservice communications.

ServiceAccountSid <String>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

InterfaceType <String>

Specifies the interface type. Each service can provide multiple service instances, each for a different purpose, and the interface defines the purpose. Available interfaces are:

SDK - for PowerShell operations

InterService - for operations between different services

Peer - for communications between services of the same type

Metadata <Citrix.EnvTest.Sdk.Metadata[]>

The collection of metadata associated with registered service instances, which are service instances held by and obtained from a central configuration service. Metadata is not stored for unregistered service instances.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-EnvTestServiceInstance
```

```
Address      : http://MyServer.com:80/Citrix/EnvTestContract
Binding      : wcf_HTTP_kerb
InterfaceType : SDK
Metadata     :
MetadataMap  :
ServiceAccount : ENG\MyAccount$
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164
ServiceGroupName : MyServiceGroup
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000
ServiceType   : EnvTest
Version      : 1
```

```
Address      : http://MyServer.com:80/Citrix/EnvTestContract/IServiceApi
Binding      : wcf_HTTP_kerb
InterfaceType : InterService
Metadata     :
MetadataMap  :
```

ServiceAccount : ENG\MyAccount  
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164  
ServiceGroupName : MyServiceGroup  
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d  
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000  
ServiceType : EnvTest  
Version : 1

Get all instances of the EnvTest Service running on the specified machine. For remote services, use the AdminAddress parameter to define the service for which the interfaces are required. If the AdminAddress parameter has not been specified for the runspace, service instances running on the local machine are returned.

# Get-EnvTestServiceStatus

Jan 04, 2017

Gets the current status of the EnvTest Service on the controller.

## Syntax

```
Get-EnvTestServiceStatus [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables the status of the EnvTest Service on the controller to be monitored. If the service has multiple data stores it will return the overall state as an aggregate of all the data store states. For example, if the site data store status is OK and the secondary data store status is DBUnconfigured then it will return DBUnconfigured.

## Related topics

[Set-EnvTestDBConnection](#)

[Test-EnvTestDBConnection](#)

[Get-EnvTestDBConnection](#)

[Get-EnvTestDBSchema](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Get-EnvTestServiceStatus command returns an object containing the status of the EnvTest Service together with extra diagnostics information.

DBUnconfigured

The EnvTest Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the EnvTest Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

#### InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the EnvTest Service schema has not been added to the database.

#### DBNotFound

The specified database could not be located with the configured connection string.

#### DBNewerVersionThanService

The version of the EnvTest Service currently in use is incompatible with the version of the EnvTest Service schema on the database. Upgrade the EnvTest Service to a more recent version.

#### DBOlderVersionThanService

The version of the EnvTest Service schema on the database is incompatible with the version of the EnvTest Service currently in use. Upgrade the database schema to a more recent version.

#### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

#### OK

The EnvTest Service is running and is connected to a database containing a valid schema.

#### Failed

The EnvTest Service has failed.

#### Unknown

(0) The service status cannot be determined.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-EnvTestServiceStatus
```

DBUnconfigured

Get the current status of the EnvTest Service.

# Get-EnvTestSuiteDefinition

Jan 04, 2017

Gets one or more test suite definitions.

## Syntax

```
Get-EnvTestSuiteDefinition [-TestSuiteId <String[]>] [-CultureName <String>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Returns a list of test suite definitions that are available from currently running components.

## Related topics

[Get-EnvTestDefinition](#)

[Get-EnvTestTask](#)

[Start-EnvTestTask](#)

[Switch-EnvTestTask](#)

[Stop-EnvTestTask](#)

[Remove-EnvTestTask](#)

[Add-EnvTestTaskMetadata](#)

[Remove-EnvTestTaskMetadata](#)

## Parameters

**-TestSuiteId**<String[]>

The id of one or more test suites.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-CultureName**<String>

The culture name in which to produce results. The culture name is in standard language/region-code format; for example "en-US".

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Input Type

System.String A test suite id. System.String[] An array of test suite ids.

Return Values

Citrix.EnvTest.Sdk.EnvTestSuiteDefinition

The definition of a test suite

Examples

----- **EXAMPLE 1** -----

```
$allTestSuiteDefinitions = Get-EnvTestSuiteDefinition
```

Retrieve all test suites.

----- **EXAMPLE 2** -----

```
$allTestSuiteDefinitionsTranslatedIntoSpanish = Get-EnvTestSuiteDefinition -CultureName es-ES
```

Retrieve all test suites with localized properties returned in Spanish.

----- **EXAMPLE 3** -----

```
$infrastructureSuiteDefinition = Get-EnvTestSuiteDefinition -TestSuiteId Infrastructure
```

Retrieve the definition of the 'Infrastructure' test suite.

# Get-EnvTestTask

Jan 04, 2017

Gets one or more EnvTestTask(s)

## Syntax

```
Get-EnvTestTask [-TaskId <Guid>] [-List] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns either the current task, a specified task, or list of tasks that are currently known to the EnvTest Service.

## Related topics

[Get-EnvTestDefinition](#)

[Get-EnvTestSuiteDefinition](#)

[Start-EnvTestTask](#)

[Switch-EnvTestTask](#)

[Stop-EnvTestTask](#)

[Remove-EnvTestTask](#)

[Add-EnvTestTaskMetadata](#)

[Remove-EnvTestTaskMetadata](#)

## Parameters

**-TaskId**<Guid>

Specifies the task identifier to be returned. This value can be retrieved from an existing task's \$task.TaskId property.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-List**<SwitchParameter>

List all running tasks, including the current one.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value | false |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Return Values

Citrix.EnvTest.Sdk.EnvTestTask

A description of a previously started task.

Examples

----- **EXAMPLE 1** -----

`$currentTask = Get-EnvTestTask`

Retrieve the current task. The current task is the most recently created task unless Switch-EnvTestTask explicitly changes it.

----- **EXAMPLE 2** -----

`$taskOfSpecifid = Get-EnvTestTask -TaskId 36C0EC52-2039-4D6E-B690-9F02F8CEFFCC`

Retrieve a fresh copy of a task object based on a known task id, which is always a Guid. This id can be retrieved from an existing task object via its \$task.TaskId property.

----- **EXAMPLE 3** -----

`$allKnownTasks = Get-EnvTestTask -List`

Retrieve the list of current tasks. This list includes any task started by the Start-EnvTestTask cmdlet since the service started that has not later been removed via Remove-EnvTestTask.

# New-EnvTestDiscoveryTargetDefinition

Jan 04, 2017

Creates a new EnvTestDiscoveryTargetDefinition object

## Syntax

```
New-EnvTestDiscoveryTargetDefinition -TestId <String> [-TargetIdType <String>] [-TargetId <String>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-EnvTestDiscoveryTargetDefinition -TestSuiteId <String> [-TargetIdType <String>] [-TargetId <String>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Creates a new EnvTestDiscoveryTargetDefinition object that can be piped into Start-EnvTestTask to define one or more targets of execution, optionally including root objects for discovery.

## Related topics

[Get-EnvTestDefinition](#)

[Get-EnvTestSuiteDefinition](#)

[Get-EnvTestTask](#)

[Start-EnvTestTask](#)

[Switch-EnvTestTask](#)

[Stop-EnvTestTask](#)

[Remove-EnvTestTask](#)

[Add-EnvTestTaskMetadata](#)

[Remove-EnvTestTaskMetadata](#)

## Parameters

**-TestId**<String>

Test identifiers. If specified, do not specify -TestSuiteId.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          | Empty |
| Accept Pipeline Input? | false |

**-TestSuiteId**<String>

Test suite identifiers. If specified, do not specify -TestId.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          | Empty |
| Accept Pipeline Input? | false |

### -TargetIdType<String>

Describes the type of corresponding object passed with -TargetId

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | Empty |
| Accept Pipeline Input? | false |

### -TargetId<String>

The Ids that object tests or test suites will target. By default, other components are queried for objects related to these.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | Empty |
| Accept Pipeline Input? | false |

### -AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

Citrix.EnvTest.Sdk.EnvTestDiscoveryTargetDefinition

Defines a target of a task

### Examples

#### ----- EXAMPLE 1 -----

```
$singleSimpleTestTaskTarget = New-EnvTestDiscoveryTargetDefinition -TestId Monitor_RegisteredWithConfigurationService
```

```
$singleSimpleTestTaskTarget | Start-EnvTestTask
```

Create a discovery target definition with a single test and no target object, then start a task based on it.

#### ----- EXAMPLE 2 -----

```
$singleSimpleTestSuiteTaskTarget = New-EnvTestDiscoveryTargetDefinition -TestSuiteId Infrastructure
```

```
$singleSimpleTestSuiteTaskTarget | Start-EnvTestTask
```

Create a discovery target definition with a single test suite and no target object, then start a task based on it.

#### ----- EXAMPLE 3 -----

```
$singleTestSuiteTaskTarget = New-EnvTestDiscoveryTargetDefinition -TestSuiteId Catalog -TargetIdType Catalog -TargetId $(Get-BrokerCatalog).Uuid
```

```
$singleTestSuiteTaskTarget | Start-EnvTestTask
```

Create a discovery target definition with a single test suite and a catalog target object, then start a task based on it.

#### ----- EXAMPLE 4 -----

```
$singleSimpleTestSuiteTaskTarget = New-EnvTestDiscoveryTargetDefinition -TestSuiteId Infrastructure
$singleTestSuiteTaskTarget = New-EnvTestDiscoveryTargetDefinition -TestSuiteId Catalog -TargetIdType Catalog -TargetId $(Get-BrokerCatalog).Uuid
@($singleSimpleTestSuiteTaskTarget, $singleTestSuiteTaskTarget) | Start-EnvTestTask
```

Create two different discovery target definitions, put them in an array, then start a task based on both.

# Remove-EnvTestServiceMetadata

Jan 04, 2017

Removes metadata from the given Service.

## Syntax

```
Remove-EnvTestServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-EnvTestServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-EnvTestServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-EnvTestServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Service.

## Related topics

[Set-EnvTestServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Service[]>

Objects to which metadata is to be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Map<PSObject>**

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name<String>**

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId<Guid>**

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-EnvTestService | % { Remove-EnvTestServiceMetadata -Map $_.MetadataMap }  
Remove all metadata from all Service objects.
```

# Remove-EnvTestTask

Jan 04, 2017

Removes from the database completed tasks for the EnvTest Service.

## Syntax

```
Remove-EnvTestTask [-TaskId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-EnvTestTask [-Task <EnvTestTask>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables completed tasks that have run within the EnvTest Service to be removed from the database.

## Related topics

[Get-EnvTestDefinition](#)

[Get-EnvTestSuiteDefinition](#)

[Get-EnvTestTask](#)

[Start-EnvTestTask](#)

[Switch-EnvTestTask](#)

[Stop-EnvTestTask](#)

[Add-EnvTestTaskMetadata](#)

[Remove-EnvTestTaskMetadata](#)

## Parameters

**-TaskId**<Guid>

Specifies the identifier of the task to be removed, retrievable from the `$task.TaskId` property.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Task**<EnvTestTask>

Specifies the task to be removed.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                |
|------------------------|----------------|
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Examples

----- **EXAMPLE 1** -----

`Remove-EnvTestTask`

Removes the current task from the EnvTest service.

----- **EXAMPLE 2** -----

`Remove-EnvTestTask -TaskId 50A4139F-2B55-4A97-A1BE-20EE4E124AA3`

Removes a task from the EnvTest service via its id, which is available from an existing task's \$task.TaskId property.

----- **EXAMPLE 3** -----

`$secondTask = $(Get-EnvTestTask -List)[1]`

`Remove-EnvTestTask -Task $secondTask`

Removes the second task in the list returned by `Get-EnvTestTask -List`.

# Remove-EnvTestTaskMetadata

Jan 04, 2017

Removes metadata from the given Task.

## Syntax

```
Remove-EnvTestTaskMetadata [-TaskTaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-EnvTestTaskMetadata [-TaskTaskId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-EnvTestTaskMetadata [-InputObject] <EnvTestTask[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Remove-EnvTestTaskMetadata [-InputObject] <EnvTestTask[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Task.

## Related topics

[Set-EnvTestTaskMetadata](#)

## Parameters

**-TaskTaskId**<Guid>

Id of the Task

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<EnvTestTask[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

### **-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with `@{"name1" = "val1"; "name2" = "val2"}`) or a string dictionary (created with `new-object "System.Collections.Generic.Dictionary[String,String]"`). The properties whose names match keys in the map will be removed.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

### **-Name**<String>

The metadata property to remove.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the `Start-LogHighLevelOperation` and `Stop-LogHighLevelOperation` cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-EnvTestTask | % { Remove-EnvTestTaskMetadata -Map $_.MetadataMap }  
Remove all metadata from all Task objects.
```



# Reset-EnvTestServiceGroupMembership

Jan 04, 2017

Reloads the access permissions and configuration service locations for the EnvTest Service.

## Syntax

```
Reset-EnvTestServiceGroupMembership [-ConfigServiceInstance] <ServiceInstance[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables you to reload EnvTest Service access permissions and configuration service locations. The Reset-EnvTestServiceGroupMembership command must be run on at least one instance of the service type (EnvTest) after installation and registration with the configuration service. Without this operation, the EnvTest services will be unable to communicate with other services in the XenDesktop deployment. When the command is run, the services are updated when additional services are added to the deployment, provided that the configuration service is not stopped. The Reset-EnvTestServiceGroupMembership command can be run again to refresh this information if automatic updates do not occur when new services are added to the deployment. If more than one configuration service instance is passed to the command, the first instance that meets the expected service type requirements is used.

## Related topics

### Parameters

#### **-ConfigServiceInstance**<ServiceInstance[]>

Specifies the configuration service instance object that represents the service instance for the type 'InterService' that references a configuration service for the deployment.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

#### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Input Type

Citrix.EnvTest.Sdk.ServiceInstance[] Service instances containing a ServiceInstance object that refers to the central configuration service interservice interface can be piped to the Reset-EnvTestServiceGroupMembership command.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoSuitableServiceInstance

None of the supplied service instance objects were suitable for resetting service group membership.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-EnvTestServiceGroupMembership
```

Reset the service group membership for a service in a deployment where the configuration service is configured and running on the same machine as the service.

----- **EXAMPLE 2** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config -AdminAddress OtherServer.example.com | Reset-EnvTestServiceGroupmembership
```

Reset the service group membership for a service in a deployment where the configuration service that is configured and running on a machine named 'OtherServer.example.com'.

# Set-EnvTestConfiguration

Jan 04, 2017

Sets the Environment Test Service's configuration options

## Syntax

```
Set-EnvTestConfiguration [-PerTypeDiscoveredObjectLimit <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Sets the Environment Test Service's configuration options and broadcasts the changes to other machines in the Site.

## Related topics

## Parameters

### **-PerTypeDiscoveredObjectLimit**<Int32>

Sets the maximum number of objects of a particular type to be explored when discovering objects during a test run.

Default: 50

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Input Type

Integer Must be greater than zero.

### Return Values

System.String

### Examples

----- **EXAMPLE 1** -----

```
Set-EnvTestConfiguration -PerTypeDiscoveredObjectLimit 100
```

Set the maximum to 100

# Set-EnvTestDBConnection

Jan 04, 2017

Configures a database connection for the EnvTest Service.

## Syntax

```
Set-EnvTestDBConnection [-DBConnection] <String> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Configures a connection to a database in which the EnvTest Service can store its state. The service will attempt to connect and start using the database immediately after the connection is configured. The database connection string is updated to the specified value regardless of whether it is valid or not. Specifying an invalid connection string prevents a service from functioning until the error is corrected.

After a connection is configured, you cannot alter it without first clearing it (by setting the connection to \$null).

You do not need to configure a database connection to use this command.

## Related topics

[Get-EnvTestServiceStatus](#)

[Get-EnvTestDBConnection](#)

[Test-EnvTestDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be used by the EnvTest Service. Passing in \$null will clear any existing database connection configured.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Force**<SwitchParameter>

If present, allows the local administrator to set the connection string to null when there are problems contacting the database or other services.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | false |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Set-EnvTestDBConnection command returns an object containing the status of the EnvTest Service together with extra diagnostics information.

DBUnconfigured

The EnvTest Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the EnvTest Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the EnvTest Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the EnvTest Service currently in use is incompatible with the version of the EnvTest Service schema on the database. Upgrade the EnvTest Service to a more recent version.

DBOlderVersionThanService

The version of the EnvTest Service schema on the database is incompatible with the version of the EnvTest Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The EnvTest Service is running and is connected to a database containing a valid schema.

Failed

The EnvTest Service has failed.

Unknown

The status of the EnvTest Service cannot be determined.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidDBConnectionString

The database connection string has an invalid format.

#### DatabaseConnectionDetailsAlreadyConfigured

There was already a database connection configured. After a configuration is set, it can only be set to \$null.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-EnvTestDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Configures a database connection string for the EnvTest Service.

----- **EXAMPLE 2** -----

```
c:\PS>Set-EnvTestDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Configures an invalid database connection string for the EnvTest Service.

# Set-EnvTestServiceMetadata

Jan 04, 2017

Adds or updates metadata on the given Service.

## Syntax

```
Set-EnvTestServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-EnvTestServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-EnvTestServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-EnvTestServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Allows you to store additional custom data against given Service objects.

## Related topics

[Remove-EnvTestServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<Service[]>

Objects to which metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |



**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Service specified. The property cannot contain any of the following characters \/:;#.\*?=<>|[]()''

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.Collections.Generic.Dictionary[String,String]

Set-EnvTestServiceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-EnvTestServiceMetadata -ServiceHostId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Key      | Value |
|----------|-------|
| ---      | ----  |
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the Service with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Set-EnvTestTaskMetadata

Jan 04, 2017

Adds or updates metadata on the given Task.

## Syntax

```
Set-EnvTestTaskMetadata [-TaskTaskId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-EnvTestTaskMetadata [-TaskTaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-EnvTestTaskMetadata [-InputObject] <EnvTestTask[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-EnvTestTaskMetadata [-InputObject] <EnvTestTask[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given Task objects.

## Related topics

[Remove-EnvTestTaskMetadata](#)

## Parameters

**-TaskTaskId**<Guid>

Id of the Task

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<EnvTestTask[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Task specified. The property cannot contain any of the following characters \/:;#.\*?=<>|[]()"

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

System.Collections.Generic.Dictionary[String,String]

Set-EnvTestTaskMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

**Notes**

If the command fails, the following errors can be returned.

**Error Codes**

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-EnvTestTaskMetadata -TaskTaskId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Key      | Value |
|----------|-------|
| ---      | ----- |
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the Task with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Start-EnvTestTask

Jan 04, 2017  
Starts a new test task.

## Syntax

```
Start-EnvTestTask -TestId <String> [-TargetIdType <String>] [-TargetId <String>] [-CultureName <String>] [-IgnoreRelatedObjects] [-RunAsynchronously] [-ExcludeNotRunTests] [-AdminAddress <String>] [<CommonParameters>]
```

```
Start-EnvTestTask -TestSuiteId <String> [-TargetIdType <String>] [-TargetId <String>] [-CultureName <String>] [-IgnoreRelatedObjects] [-RunAsynchronously] [-ExcludeNotRunTests] [-AdminAddress <String>] [<CommonParameters>]
```

```
Start-EnvTestTask -InputObject <PSObject[]> [-CultureName <String>] [-IgnoreRelatedObjects] [-RunAsynchronously] [-ExcludeNotRunTests] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Starts a new test task based on a set of criteria provided via parameters or piped input and either waits for the tests to run or returns immediately depending on how it is called. When running a test suite and providing a target object for that suite, the service will discover related objects by default, but this behavior may be disabled if desired.

## Related topics

- [Get-EnvTestDefinition](#)
- [Get-EnvTestSuiteDefinition](#)
- [Get-EnvTestTask](#)
- [New-EnvTestDiscoveryTargetDefinition](#)
- [Switch-EnvTestTask](#)
- [Stop-EnvTestTask](#)
- [Remove-EnvTestTask](#)
- [Add-EnvTestTaskMetadata](#)
- [Remove-EnvTestTaskMetadata](#)

## Parameters

**-TestId**<String>

Test identifiers. If specified, do not specify -TestSuiteId.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          | Empty |
| Accept Pipeline Input? | false |

**-TestSuiteId**<String>

Test suite identifiers. If specified, do not specify -TestId.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          | Empty |
| Accept Pipeline Input? | false |

**-InputObject**<PSObject[]>

One or more test targets defining the task, see Input Types for details about what kind of objects are permissible. Any valid object passed to this parameter may also be piped into this cmdlet.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          | Empty          |
| Accept Pipeline Input? | true (ByValue) |

**-TargetIdType**<String>

Describes the type of corresponding object passed with -TargetId

|           |       |
|-----------|-------|
| Required? | false |
|-----------|-------|



|                        |       |
|------------------------|-------|
| Default Value          | Empty |
| Accept Pipeline Input? | false |

**-TargetId**<String>

The Ids that object tests or tests suites will target. By default, other components are queried for objects related to these.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | Empty |
| Accept Pipeline Input? | false |

**-CultureName**<String>

The culture name in which to produce results. The culture name is in standard language/region-code format; for example "en-US".

|                        |                                    |
|------------------------|------------------------------------|
| Required?              | false                              |
| Default Value          | The current user interface culture |
| Accept Pipeline Input? | false                              |

**-IgnoreRelatedObjects**<SwitchParameter>

Do not ask other components for objects related to a specified target.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-RunAsynchronously**<SwitchParameter>

Do not wait for the test run to complete, return immediately.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-ExcludeNotRunTests**<SwitchParameter>

If set, tests that are not run because no object matching their requirements is found are NOT included in test counts and results.

|                        |                               |
|------------------------|-------------------------------|
| Required?              | false                         |
| Default Value          | False (include Not Run tests) |
| Accept Pipeline Input? | false                         |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Input Type

Citrix.EnvTest.Sdk.EnvTestDiscoveryTargetDefinition A single EnvTestDiscoveryTargetDefinition can be specified to target one test or test suite. Citrix.EnvTest.Sdk.EnvTestDiscoveryTargetDefinition[] An array of EnvTestDiscoveryTargetDefinition(s) can be specified to target any combination of tests and/or test suites. PSCustomObject A single PSCustomObject with properties matching the required EnvTestDiscoveryTargetDefinition properties can be specified to target one test or test suite. PSCustomObject[] An array of PSCustomObject(s) with properties matching the required EnvTestDiscoveryTargetDefinition properties can be specified to target any combination of tests and/or test suites.

#### Return Values

Citrix.EnvTest.Sdk.EnvTestTask

The newly started task.

#### Examples

##### ----- EXAMPLE 1 -----

```
$singleSimpleTestTask = Start-EnvTestTask -TestId Monitor_RegisteredWithConfigurationService
```

Create and start a task with a single test and no target object.

##### ----- EXAMPLE 2 -----

```
$singleSimpleTestTaskInSpanish = Start-EnvTestTask -TestId Monitor_RegisteredWithConfigurationService -CultureName es-ES
```

Create and start a task with a single test and no target object, with localized properties translated into Spanish.

##### ----- EXAMPLE 3 -----

```
$singleSimpleTestSuiteTask = Start-EnvTestTask -TestSuiteId Infrastructure
```

Create and start a task with a single test suite and no target object.

##### ----- EXAMPLE 4 -----

```
$singleTestSuiteTask = Start-EnvTestTask -TestSuiteId Catalog -TargetIdType Catalog -TargetId $(Get-BrokerCatalog).Uuid
```

Create and start a task with a single test suite and a catalog target object.

##### ----- EXAMPLE 5 -----

```
$singleTestSuiteTask = Start-EnvTestTask -TestSuiteId Catalog -TargetIdType Catalog -TargetId $(Get-BrokerCatalog).Uuid -RunAsynchronously
```

Create and start a task with a single test suite and a catalog target object, and return immediately not waiting for the tests to complete.

##### ----- EXAMPLE 6 -----

```
$adAccountPool = Get-AcctIdentityPool
$singleTestTaskWithNoObjectDiscovery = StartEnvTestTask -IgnoreRelatedObjects -TestId ADIdentity_IdentityPool_ValidatePoolsUnlocked -TargetIdType IdentityPool -TargetId $adAccountPool.
```

Create and start a task with a single test, a target object for that test, and no object discovery based on that target.

##### ----- EXAMPLE 7 -----

```
$singleSimpleTestSuiteTaskTarget = New-EnvTestDiscoveryTargetDefinition -TestSuiteId Infrastructure
$singleTestSuiteTaskTarget = New-EnvTestDiscoveryTargetDefinition -TestSuiteId Catalog -TargetIdType Catalog -TargetId $(Get-BrokerCatalog).Uuid
$inputObjects = @($singleSimpleTestSuiteTaskTarget, $singleTestSuiteTaskTarget)
Start-EnvTestTask -InputObject $inputObjects
```

Create two different discovery target definitions, put them in an array, then start a task based on both via -InputObject.

# Stop-EnvTestTask

Jan 04, 2017

Stops a still running task from completing.

## Syntax

```
Stop-EnvTestTask [-TaskId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Stop-EnvTestTask [-Task <EnvTestTask>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Stops a still running task from completing. A task may still be retrieved via `Get-EnvTestTask` until `Remove-EnvTestTask` is called with its task id.

## Related topics

[Get-EnvTestDefinition](#)

[Get-EnvTestSuiteDefinition](#)

[Get-EnvTestTask](#)

[New-EnvTestTask](#)

[Start-EnvTestTask](#)

[Switch-EnvTestTask](#)

[Remove-EnvTestTask](#)

[Add-EnvTestTaskMetadata](#)

[Remove-EnvTestTaskMetadata](#)

## Parameters

**-TaskId**<Guid>

The id of the task to stop, retrieveable from the `$task.TaskId` property.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Task**<EnvTestTask>

An `EnvTestTask` representing the task to stop

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Examples

----- **EXAMPLE 1** -----

Stop-EnvTestTask  
 Stops the current task from completing if it is still running.

----- **EXAMPLE 2** -----

Stop-EnvTestTask -TestId 50A4139F-2B55-4A97-A1BE-20EE4E124AA3  
 Stops a task from completing via its id, which is available from an existing task's \$task.TaskId property.

----- **EXAMPLE 3** -----

```
$secondTask = $(Get-EnvTestTask -List)[1]
Stop-EnvTestTask -Task $secondTask
```

Stops the second task in the list returned by Get-EnvTestTask -List.

# Switch-EnvTestTask

Jan 04, 2017

Sets the current task that will be returned by a call to Get-EnvTestTask with no parameters.

## Syntax

```
Switch-EnvTestTask [-TaskId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Switch-EnvTestTask [-Task <EnvTestTask>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Sets the current task that will be returned by a call to Get-EnvTestTask with no parameters.

## Related topics

[Get-EnvTestDefinition](#)

[Get-EnvTestSuiteDefinition](#)

[Get-EnvTestTask](#)

[New-EnvTestTask](#)

[Start-EnvTestTask](#)

[Stop-EnvTestTask](#)

[Remove-EnvTestTask](#)

[Add-EnvTestTaskMetadata](#)

[Remove-EnvTestTaskMetadata](#)

## Parameters

**-TaskId**<Guid>

Specifies the identifier of the task to be made current.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Task**<EnvTestTask>

The task object to be made current, retrieveable from the \$task.TaskId property.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Input Type**

System.Management.Automation.PSObject Objects containing the TaskId parameter can be piped to the Remove-EnvTestTask command.

**Examples**

----- **EXAMPLE 1** -----

```
Switch-EnvTestTask -TaskId 50A4139F-2B55-4A97-A1BE-20EE4E124AA3
```

Switches the current task to another via its id, which is available from an existing task's Stask.TaskId property.

----- **EXAMPLE 2** -----

```
$secondTask = $(Get-EnvTestTask -List)[1]
Switch-EnvTestTask -Task $switchTask
```

Switches the current task to the second in the list returned by Get-EnvTestTask -List.

# Test-EnvTestDBConnection

Jan 04, 2017

Tests a database connection for the EnvTest Service.

## Syntax

```
Test-EnvTestDBConnection [-DBConnection] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Tests a connection to the database in which the EnvTest Service can store its state. The service will attempt to connect to the database without affecting the current connection to the database.

You do not have to clear the connection to use this command.

## Related topics

[Get-EnvTestServiceStatus](#)

[Get-EnvTestDBConnection](#)

[Set-EnvTestDBConnection](#)

## Parameters

### **-DBConnection**<String>

Specifies the database connection string to be tested by the EnvTest Service.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

### Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Test-EnvTestDBConnection command returns an object containing the status of the EnvTest Service if the connection string of the specified data store were to be set to the string being tested, together with extra diagnostics information for the specified connection string.

### DBRejectedConnection

The database rejected the logon attempt from the EnvTest Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

### InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the EnvTest Service schema has not been added to the database.

### DBNotFound

The specified database could not be located with the configured connection string.

### DBNewerVersionThanService

The version of the EnvTest Service currently in use is incompatible with the version of the EnvTest Service schema on the database. Upgrade the EnvTest Service to a more recent version.

### DBOlderVersionThanService

The version of the EnvTest Service schema on the database is incompatible with the version of the EnvTest Service currently in use. Upgrade the database schema to a more recent version.

### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

### OK

The Set-EnvTestDBConnection command would succeed if it were executed with the supplied connection string.

### Failed

The EnvTest Service has failed.

### Unknown

The status of the EnvTest Service cannot be determined.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

### InvalidDBConnectionString

The database connection string has an invalid format.

### PermissionDenied

You do not have permission to execute this command.



#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
c:\PS>Test-EnvTestDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Tests a database connection string for the EnvTest Service.

##### ----- EXAMPLE 2 -----

```
c:\PS>Test-EnvTestDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Tests an invalid database connection string for the EnvTest Service.

# Citrix.Host.Admin.V2

Jan 04, 2017  
Overview

| Name                          | Description                                                               |
|-------------------------------|---------------------------------------------------------------------------|
| <a href="#">HypHostSnapin</a> | The Host Service PowerShell snap-in provides administrative functions for |
| <a href="#">Hyp Filtering</a> | Describes the common filtering options for XenDesktop cmdlets.            |

## Cmdlets

| Name                                                | Description                                                                                                                                             |
|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Add-HypHostingUnitMetadata</a>          | Adds metadata on the given HostingUnit.                                                                                                                 |
| <a href="#">Add-HypHostingUnitNetwork</a>           | Makes additional hypervisor networks available for use in a HostingUnit.                                                                                |
| <a href="#">Add-HypHostingUnitStorage</a>           | Adds storage locations to a hosting unit.                                                                                                               |
| <a href="#">Add-HypHypervisorConnectionAddress</a>  | Add a connection address to a hypervisor connection.                                                                                                    |
| <a href="#">Add-HypHypervisorConnectionMetadata</a> | Adds metadata on the given HypervisorConnection.                                                                                                        |
| <a href="#">Add-HypHypervisorConnectionScope</a>    | Add the specified HypervisorConnection(s) to the given scope(s).                                                                                        |
| <a href="#">Add-HypMetadata</a>                     | Adds metadata to a hypervisor connection or a hosting unit.                                                                                             |
| <a href="#">Get-HypConfigurationDataForItem</a>     | Retrieves the configuration data for an item in the Host Service provider path. Note: For this release, only VM items are supported for this operation. |
| <a href="#">Get-HypConfigurationObjectForItem</a>   | Retrieves the configuration data for an item in the Host Service provider path. Note: For this release, only VM items are supported for this operation. |
| <a href="#">Get-HypConnectionRegion</a>             | Enumerates the regions of a hypervisor connection that are based on cloud technology.                                                                   |
| <a href="#">Get-HypDBConnection</a>                 | Gets the database string for the specified data store used by the Host Service.                                                                         |

| Name                                  | Description                                                                               |
|---------------------------------------|-------------------------------------------------------------------------------------------|
| Get-HypDBSchema                       | Gets a script that creates the Host Service database schema for the specified data store. |
| Get-HypDBVersionChangeScript          | Gets a script that updates the Host Service database schema.                              |
| Get-HypHypervisorPlugin               | Gets the available hypervisor types.                                                      |
| Get-HypInstalledDBVersion             | Gets a list of all available database schema versions for the Host Service.               |
| Get-HypScopedObject                   | Gets the details of the scoped objects for the Host Service.                              |
| Get-HypService                        | Gets the service record entries for the Host Service.                                     |
| Get-HypServiceAddedCapability         | Gets any added capabilities for the Host Service on the controller.                       |
| Get-HypServiceInstance                | Gets the service instance entries for the Host Service.                                   |
| Get-HypServiceStatus                  | Gets the current status of the Host Service on the controller.                            |
| Get-HypVMMacAddress                   | Retrieves a list the MAC addresses for the VMs in the specified connection.               |
| Get-HypVolumeServiceConfiguration     | Gets instances of the VolumeServiceConfiguration that are configured for this site.       |
| Get-HypXenServerAddress               | Gets all the available addresses for a XenServer hypervisor connection.                   |
| Grant-HypSecurityGroupEgress          | Adds an egress rule to a security group.                                                  |
| Grant-HypSecurityGroupIngress         | Adds an ingress rule to a security group.                                                 |
| New-HypVMSnapshot                     | Creates a new snapshot for the specified VM item path.                                    |
| Remove-HypHostingUnitMetadata         | Removes metadata from the given HostingUnit.                                              |
| Remove-HypHostingUnitNetwork          | Removes networks from a hosting unit.                                                     |
| Remove-HypHostingUnitStorage          | Removes storage from a hosting unit.                                                      |
| Remove-HypHypervisorConnectionAddress | Removes addresses from the list of available connection addresses.                        |

| Remove-<br>Name<br>HypHypervisorConnectionMetadata | Removes metadata from the given HypervisorConnection.<br>Description                        |
|----------------------------------------------------|---------------------------------------------------------------------------------------------|
| Remove-<br>HypHypervisorConnectionScope            | Remove the specified HypervisorConnection(s) from the given scope(s).                       |
| Remove-HypMetadata                                 | Removes metadata from a hypervisor connection or hosting unit.                              |
| Remove-HypServiceMetadata                          | Removes metadata from the given Service.                                                    |
| Reset-HypServiceGroupMembership                    | Reloads the access permissions and configuration service locations for the Host Service.    |
| Revoke-HypSecurityGroupEgress                      | Removes an egress rule from a security group.                                               |
| Revoke-HypSecurityGroupIngress                     | Removes an ingress rule from a security group.                                              |
| Set-HypAdminConnection                             | Set the controller to be used by the cmdlets that form the Host service PowerShell snap-in. |
| Set-HypDBConnection                                | Configures a database connection for the Host Service.                                      |
| Set-HypHostingUnitMetadata                         | Adds or updates metadata on the given HostingUnit.                                          |
| Set-HypHostingUnitStorage                          | Sets options for a storage location on a hosting unit.                                      |
| Set-HypHypervisorConnectionMetadata                | Adds or updates metadata on the given HypervisorConnection.                                 |
| Set-HypServiceMetadata                             | Adds or updates metadata on the given Service.                                              |
| Set-HypVolumeServiceConfiguration                  | Applies a change to one of the VolumeServiceConfiguration instances in the site.            |
| Start-HypVM                                        | Starts a VM.                                                                                |
| Stop-HypVM                                         | Stops a VM by issuing a Shutdown request                                                    |
| Test-HypDBConnection                               | Tests a database connection for the Host Service.                                           |
| Test-HypHostingUnitNameAvailable                   | Checks to ensure that the proposed name for a hosting unit is unused.                       |
| Test-                                              | Checks to ensure that the proposed name for a hypervisor connection is                      |

| Name                              | Description                                                                                                    |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------|
| HypervisorConnectionNameAvailable | unused.                                                                                                        |
| Update-HypervisorConnection       | Requests the host service to update the connection properties that depend on the version of hypervisor in use. |

# about\_HypHostSnapin

Jan 04, 2017

## TOPIC

about\_HypHostSnapin

## SHORT DESCRIPTION

The Host Service PowerShell snap-in provides administrative functions for the Host Service.

## COMMAND PREFIX

All commands in this snap-in have 'Hyp' in their name.

## LONG DESCRIPTION

The Host Service PowerShell snap-in enables both local and remote administration of the Host Service. It lets you configure XenDesktop deployments to make use of hypervisors, networks, and storage, and enables browsing of their contents using the Host PowerShell provider (Citrix.HypervisorProvider).

The provider creates a default PSDrive with a drive identifier of 'XDHyp'. This drive provides two root folders:

### HostingUnits Folder

Contains a list of all the hosting units that have been defined using the new-Item provider command.

Hosting units define not only a hypervisor connection, but also reference networks and storage. Hosting units are used by the Machine Creation Service to provide the information required to create and manage virtual machines that can be used by other services. A hosting unit references a root path. This is a specific point in the provider connection tree. The hosting unit is constrained to provide only items below this point in the tree. This restricts the locations that the Machine Creation Service can use to create virtual machines and the networks and storage that can be used.

### Connections Folder

Contains a list of all the hosting unit connections that are defined using the new-Item provider command.

Change directory to a specific connection and use the dir/Get-ChildItem command to list all of the infrastructure (such as folders, hypervisors, networks, storage, and virtual machines) that is available in the hosting unit to which the connection refers. The paths to these items

are used as the input to commands in the Host Service and Machine Creation Service snap-ins.

The contents of the Hosting Unit and Connection folders reflect the content and structure of the hypervisor to which they refer. The item extensions reflect the object type for each item. Not all item types are appropriate for all hypervisor types. Possible item types are:

- Virtual Machine (.vm)
- Snapshot (.snapshot)
- Cluster (.cluster)
- Host (.host)
- HostGroup (.hostgroup)
- DataCenter (.datacenter)
- Folder (.folder)
- ResourcePool (.resourcepool)
- ComputeResource (.computeresource)

When used with a path that refers to the Host provider (with a default drive of 'XDHyp:'), the Host provider extends the standard New-Item, Get-Item, Get-ChildItem, Remove-Item, Rename-Item, and Set-Item commands as described below. For more information about the basic behavior of these commands, see the help for the command. The Move-Item and Copy-Item commands are not supported for the Host provider.

New-Item

The following parameters are available when using New-Item in the Connections directory (or a path that refers to the directory). The Credential parameter is not supported.

-Name

Specifies the name of the connection, which must not contain any of the following characters: \;#.\*?=<>|[](){}|. The Name parameter is optional but, if not included, the connection name must be specified as part of the Path parameter.

-HypervisorAddress <String[]>

Specifies an array of addresses that can be used to contact the required hypervisor. All the addresses are considered equivalent, that is, all of the addresses provide access to the same virtual machines, snapshots, network, and storage.

-ConnectionType <ConnectionType>

Specifies the type of hypervisor that the connection is for. Supported hypervisor types are:

XenServer

SCVMM (Microsoft Hyper-V)  
vCenter (VMware vSphere/ESX)  
Custom

**-PluginId <String>**

Specifies the class name for the Citrix Managed Machine library that is used to access the hypervisor. You can obtain this list using the `Get-HypHypervisorPlugin` command. The `PluginId` parameter must be specified if the `ConnectionType` is set to 'Custom'.

**-UserName <String>, -Password <String>, -SecurePassword <SecureString>**

Specifies the credentials for the connection to the hypervisor. You can specify the password as either `Password` or `SecurePassword`, but not both. The `UserName` parameter, and either `Password` or `SecurePassword`, specify the same information as the `HypervisorCredential` parameter, so use of one precludes use of the other.

**-HypervisorCredential <PSCredential>**

Specifies credentials for the connection to the hypervisor. The `HypervisorCredential` parameter specifies the same information as `UserName` and either `Password` or `SecurePassword`, so use of one precludes use of the other.

**-Persist**

Specifies that the connection details are persistent. If this parameter is not included, the connection is held only for the duration of the current runspace and `PSDrive` combination. Only persistent connection items are visible to administrators in other runspaces and `PSDrives`. Hosting units cannot be created from connections that are not persistent.

**-AdminAddress <String>**

Specifies the address of the Host Service that the command communicates with. After it is set, this address is used for all commands in the Host Service PowerShell snap-in. If this parameter is not included or set by another command, or if `Set-HypAdminConnection` has not been used, the command attempts to use a local Host Service.

**-LoggingId <String>**

Specifies the identifier of the high-level operation that this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the `Start-LogHighLevelOperation` and `Stop-LogHighLevelOperation` cmdlets.

**-Scopes <String[]>**



Specifies the list of administrative scopes that the new connection will be a part of. The scopes control which administrators are able to work with the connection.

The following parameters are available when using `New-Item` in the `HostingUnits` directory (or a path that refers to the directory).

`-Name`

Specifies the name of the hosting unit, which must not contain any of the following characters: `\;#.*?=<>|[](){}.` The `Name` parameter is optional but, if not included, the hosting unit name must be provided as part of the `Path` parameter.

`-HypervisorConnectionName <String>`

Specifies the name of the connection that the hosting unit uses. To create a hosting unit, the referenced connection must be persistent and the `ConnectionType` must not be set to 'Custom'.

`-HypervisorConnectionUid <Guid>`

Specifies the unique identifier of the connection that the hosting unit uses. To create a hosting unit, the referenced connection must be persistent and the `ConnectionType` must not be set to 'Custom'.

`-RootPath <String>`

Specifies the location in a connection that is used as the starting reference for the hosting unit. The path must point to an item in the connection that is marked as a `SymLink`. The root of a `XenServer` connection is a special case that is also considered a `SymLink`. If this parameter is not included, the current location in the provider is used.

`-NetworkPath <String>`

Specifies the path in a connection to the network item that is used when the `Machine Creation Service` creates new virtual machines.

`-StoragePath <String>`

Specifies one or more paths in a connection to storage items that are used when the `Machine Creation Service` creates new virtual machines. After they are set, you can modify storage paths using the `Add-HypHostingUnitStorage` and `Remove-HypHostingUnitStorage` commands. If the connection is based on cloud infrastructure, storage items are typically not available, in which case this parameter can be omitted.

`-PersonalVdiskStoragePath <String>`

Specifies one or more paths in a connection to storage items

that are used when the Machine Creation Service creates disks for the virtual machines. After they are set, you can modify storage paths using the Add-HypHostingUnitStorage and Remove-HypHostingUnitStorage commands.

**-NoVmTagging**

Specifies that new virtual machines are not tagged with metadata from the hypervisor. By default, all virtual machines created by the Machine Creation Service are tagged to show they are created by XenDesktop. These tags are used by the provider to restrict the list of virtual machines displayed when viewing the content of the Connections or HostingUnit paths. If this parameter is not included, all virtual machines are displayed at all times.

**-AdminAddress <String>**

Specifies the address of the Host Service that the command will communicate with. After it is set, this address is used for all commands in the Host Service PowerShell snap-in. If this parameter is not included or set by another command, or if Set-HypAdminConnection has not been used, the command attempts to use a local Host Service.

**-UseLocalStorageCaching**

When Get-HypServiceAddedCapability indicates that the LocalStorageCaching feature is available, use this parameter to specify that the virtual machines created for this hosting unit will use local storage caching for their disk images.

**-GpuGroup**

Specifies a path to a GPU Group in a connection that will be used when

Machine Creation Services creates VMs. Only a single GPU Group is supported and the value is immutable.

**-LoggingId <String>**

Specifies the identifier of the high-level operation that this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

The following parameters are available when using New-Item relative to a connection or hosting unit.

**-ItemType <string>**

Specifies the type of item to create when invoked relative to a connection or hosting unit. Supported ItemType values are:

SecurityGroup (cloud only)

-Description

Specifies a description for the security group.

-VpcId

Specifies a VPC ID for the security group.

Get-Item

Get-ChildItem (alias dir)

Rename-Item

Remove-Item

The following additional parameters are available.

AdminAddress

Specifies the address of the Host Service that the command communicates with. After it is set, this address is used for all commands in the Host Service PowerShell snap-in. If this parameter is not included or set by another command, or if Set-HypAdminConnection has not been used, the command attempts to use a local Host Service.

-LoggingId <String>

Specifies the identifier of the high-level operation that this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

You can specify the Path parameter by name or by using the unique identifier for the connection or hosting unit enclosed in braces.

For example:

```
-Path XDHyp:\{1233-3213ACDF-12323}
```

The Filter parameter is not supported. Virtual machines created by the Machine Creation Service are returned only if the -Force parameter is used or if virtual machine tagging was disabled on the hosting unit with the NoVmTagging parameter when the VMs were created.

## Set-Item

The following parameters are available when using Set-Item in the Connections directory.

**-UserName <String>, -Password <String>, -SecurePassword <SecureString>**

Specifies the credentials for the connection to the hypervisor. The password can be given as either Password or SecurePassword, but not both. The UserName parameter and either Password or SecurePassword specify the same information as the HypervisorCredential parameter, so use of one precludes use of the other.

**-HypervisorCredential <PSCredential>**

Specifies credentials for the connection to the hypervisor. The HypervisorCredential parameter specifies the same information as the UserName parameter and either Password or SecurePassword, so use of one precludes use of the other.

**-HypervisorAddress <string[]>**

Specifies the addresses of the hypervisor that this connection represents. This replaces all existing addresses.

**-LoggingId <String>**

Specifies the identifier of the high-level operation that this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

**-MaintenanceMode <Boolean>**

Places the connection into maintenance mode which disables all communication between XenDesktop and the Hypervisor. Use this mode when making changes to the hypervisor; for instance, if the password for the access account needs to be changed.

**-AdminAddress <String>**

Specifies the address of the Host Service that the command communicates with. After it is set, this address is used for all commands in the Host Service PowerShell snap-in. If this parameter is not included or set by another command, or if Set-HypAdminConnection has not been used, the command tries to use a local Host Service.

The Credential parameter is not supported. Addresses can be modified with the Add-HypHypervisorConnectionAddress and Remove-HypHypervisorConnectionAddress commands. Metadata can be modified with the Add-HypMetadata and Remove-HypMetadata commands.

The following parameters are available when using New-Item in the

HostingUnits directory.

**-Name**

Specifies the name of the hosting unit, which must not contain any of the following characters: \;#.\*?=<>|[](){}". The Name parameter is optional but, if not included, the hosting unit name must be provided as part of the Path parameter.

**-NetworkPath <String>**

Specifies the path in a connection to the network item that is used when the Machine Creation Service creates new virtual machines.

**-NoVmTagging**

Specifies that new virtual machines are not tagged with metadata from the hypervisor. By default, all virtual machines created by the Machine Creation Service are tagged to show they are created by XenDesktop. These tags are used by the provider to restrict the list of virtual machines displayed when viewing the content of the Connections or HostingUnit paths. If this parameter is not included, all virtual machines are displayed at all times.

**-AdminAddress <String>**

Specifies the address of the Host Service that the command communicates with. After it is set, this address is used for all commands in the Host Service PowerShell snap-in. If this parameter is not included or set by another command, or if Set-HypAdminConnection has not been used, the command attempts to use a local Host Service.

**-UseLocalStorageCaching <bool>**

When Get-HypServiceAddedCapability indicates that the LocalStorageCaching feature is available, use this parameter to specify that the virtual machines created for this hosting unit will use local storage caching for their disk images.

**-LoggingId <String>**

Specifies the identifier of the high-level operation that this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Storage can be modified with the Add-HypHostingUnitStorage and Remove-HypHostingUnitStorage commands. Metadata can be modified with the Add-HypMetadata and Remove-HypMetadata commands.

The item types returned when these commands are used in the Host Service provider are defined in the Object Definitions section below.

## EXAMPLES

### To Create a Persistent Hypervisor Connection

```
new-item -path "xdhyp:\Connections" -Name MyConn -ConnectionType
  XenServer -HypervisorAddress "http:\address" -UserName user
  -Password password -Persist
```

```
PSPPath:          Citrix.HostingUnitService.Admin.V1.0\
                  Citrix.Hypervisor::XDHyp:\Connections\MyConn
PSParentPath:     Citrix.HostingUnitService.Admin.V1.0\
                  Citrix.Hypervisor::XDHyp:\Connections
PSChildName:      MyConn
PSDrive:          XDHyp
PSProvider:       Citrix.HostingUnitService.Admin.V1.0\
                  Citrix.Hypervisor
PSIsContainer:    True
HypervisorConnectionUid: 04e6daa2-5cbd-4491-b70c-6daf733ee82a
HypervisorConnectionName: MyConn
ConnectionType:   XenServer
HypervisorAddress: {http:\address}
UserName:         user
Persistent:       True
PluginId:         XenFactory
SupportsPvsVMs:  True
Revision:         1b0b0d02-bc1b-49d8-b2b0-be6fb7f150ad
MaintenanceMode: False
Metadata:         {MaxAbsoluteActiveActions = 100,
                  MaxAbsoluteNewActionsPerMinute = 100,
                  MaxPowerActionsPercentageOfDesktops = 20}
```

### To Create a Hosting Unit

```
new-item -Path "xdhyp:\HostingUnits"
  -Name MyHU
  -HypervisorConnectionName MyConn
  -RootPath XDHYP:\Connections\MyConn
  -NetworkPath XDHYP:\Connections\MyConn\Network 0.network
  -StoragePath XDHYP:\Connections\MyConn\Local storage on
  myXenServer.storage
```

```
PSPPath:          Citrix.HostingUnitService.Admin.V1.0\
                  Citrix.Hypervisor::XDHyp:\HostingUnits\MyHU
PSParentPath:     Citrix.HostingUnitService.Admin.V1.0\
                  Citrix.Hypervisor::XDHyp:\HostingUnits
```

```
PSChildName:    MyHU
PSDrive:        XDHyp
PSPProvider:    Citrix.HostingUnitService.Admin.V1.0\
                Citrix.Hypervisor
PSIsContainer:  True
HostingUnitUid: df91f886-1141-4280-bd59-2ee260a4df79
HostingUnitName: MyHU
HypervisorConnection: MyConn
RootPath:       /
RootId:
NetworkPath:    /Network 0.network
NetworkId:      ab47080b-ca15-771a-c8dc-6ad9650158f1
Storage:        {/Local storage on myXenServer.storage}
VMTaggingEnabled: True
Metadata:       {}
```

Relative paths can be used for all parameters.

## OBJECT DEFINITIONS

### Citrix.XDInterServiceTypes.HypervisorConnection

The hypervisor connection object is returned when a connection item is located. This item has the following parameters.

HypervisorConnectionUid <Guid>

Specifies the unique identifier for the connection item.

HypervisorConnectionName <String>

Specifies the name of the connection item.

ConnectionType <Citrix.XDInterServiceTypes.ConnectionType>

Specifies the type of hypervisor that the connection is for.

Supported hypervisor types are:

- XenServer
- SCVMM (Microsoft Hyper-V)
- vCenter (VMware vSphere/ESX)
- Custom

HypervisorAddress <String[]>

Specifies the addresses that can be used to contact the required hypervisor.

Username <String>

Specifies the administrator user name for the connection to the hypervisor (from the user name and password given as administrator credentials when

setting up this connection)

Persistent <Boolean>

Specifies whether or not the connection is persistent.

PluginId <String>

Specifies the Citrix Managed Machine class identifier for the hypervisor.

SupportsPvsVM <Boolean>

Specifies whether or not the connection can be used as part of a hosting unit. If this parameter is set to 'True', a hosting unit referencing this connection can be created.

Revision <Guid>

A unique identifier that is changed every time any properties of the connection are changed.

MaintenanceMode <Boolean>

Specifies whether or not the connection is currently in maintenance mode.

Metadata <Citrix.XDInterServiceTypes.Metadata[]>

The collection of metadata associated with the connection.

#### Citrix.XDInterServiceTypes.HostingUnit

The hosting unit object is returned when a hosting unit item is located. This item has the following parameters.

HostingUnitName <String>

Specifies the name of the hosting unit.

HostingUnitUid <Guid>

Specifies the unique identifier for the hosting unit.

HypervisorConnection <Citrix.XDInterServiceTypes.HypervisorConnection>

Specifies the hypervisor connection item that the hosting unit references.

NetworkId <String>

Specifies the unique identifier for the network in the hypervisor context that the hosting unit references.

NetworkPath <String>

Specifies the path to the network in the Host Service provider.



RootId <String>

Specifies the unique identifier for the root connection item in the hypervisor context that the hosting unit references.

RootPath <String>

Specifies the path to the root of the hosting unit from within the Host Service provider.

Storage <Citrix.XDInterServiceTypes.Storage[]>

Specifies the collection of storage objects that are defined for use as part of the hosting unit. This object contains the following parameters.

StorageID <String>

Specifies the identifier for the storage in the hypervisor context.

StoragePath <String>

Specifies the path to the storage in the Host Service provider.

VMTaggingEnabled <Boolean>

Specifies whether or not virtual machine metadata is used to tag virtual machines created by XenDesktop.

Metadata <Citrix.XDInterServiceTypes.Metadata[]>

Specifies the collection of metadata associated with the hosting unit.

Citrix.HostingUnitService.Sdk.HypervisorObject

The hypervisor object is returned when any item is located from within a Connection or HostingUnit folder. This item has the following parameters.

AdditionalData <Dictionary<String,String>

Stores extra untyped data for the item. This dictionary contains different information for each item type.

For Storage Items

Key = StorageType

Values = Shared or Local

For VM Items

Key = PowerState Values = PoweredOn,

PoweredOff,

Suspended,  
TransitioningToOn,  
TransitioningToOff,  
Suspending,  
Resuming,  
Unknown,  
Error

Name <String>  
Specifies the name of the item.

FullName <String>  
Specifies the name with the appropriate file extension.

ObjectPath <String>  
Specifies the relative path to the item from the root of the connection of which the item is part.

FullPath <String>  
Specifies the absolute path to the item in the Citrix.Hypervisor provider.

Id <String>  
Specifies the identity of the item in the hypervisor context.

IsContainer <Boolean>  
Specifies whether or not the item can contain other items.

IsSymLink <Boolean>  
Specifies whether or not the item can be used as a RootPath of a hosting unit.

ObjectType <Citrix.HostingUnitService.Sdk.NodeType>  
Specifies the item type.

## ERROR CODES

The provider commands can return the following error codes.

New-Item

ConnectionNameOrUidInvalid  
The name or unique identifier of the hypervisor connection

specified for the hosting unit is invalid.

#### HostingUnitRootPathInvalid

The root path specified for the hosting unit is invalid. The root path must be a valid item in the hypervisor tree and a SymLink.

#### HostingUnitNetworkPathInvalid

The network path specified for the hosting unit is invalid. The network path must be a valid item in the hypervisor tree and relative to the root path.

#### HostingUnitStoragePathInvalid

The storage path specified for the hosting unit is invalid. The storage path must be a valid item in the hypervisor tree and relative to the root path.

#### HostingUnitDuplicateObjectExists

A hosting unit object with the same name already exists. Hosting unit names must be unique.

#### HostingUnitStorageDuplicateObjectExists

A hosting unit storage object with the same storage path already exists for this hosting unit. There can be only one object for each combination of hosting unit and storage path.

#### HypervisorNotContactable

The hypervisor could not be contacted at the supplied address, which is either invalid or unreachable.

#### HypervisorAddressInvalidFormat

The address is not in a valid form for the specified hypervisor type.

#### ConnectionAddressInvalid

The specified address is invalid and does not belong to the same pool.

#### HypervisorConnectionDuplicateObjectExists

A hypervisor connection object with the same name already exists. Hypervisor connection names must be unique.

#### HypervisorConnectionAddressDuplicateObjectExists

A hypervisor connection address object with the same address already exists for this hypervisor connection. There can be only one object for each combination of hypervisor connection and address.

#### HypervisorConnectionForHostingUnitsVirtual

The specified hypervisor connection for the hosting unit is a virtual

non-persistent connection. A persistent connection is required when creating a hosting unit.

#### InputNameInvalid

The name specified for the hosting unit or hypervisor connection is invalid because it contains one or more of the following characters:

V;#:.\*?=<>|[](){}.

#### InputPathInvalid

The specified path is invalid because it is not in one of the following formats:

XDHyp:\Connections\

XDHyp:\Connections\{Guid}

#### ExceptionThrown

An unexpected error occurred.

#### DatabaseError

There was a problem communicating with the database.

#### CommunicationError

There was a problem communicating with the remote service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ScopeNotFound

One or more of the scopes nominated for the new connection do not exist.

#### CannotCreateSecurityGroupHere

Security groups can only be created in a virtual private cloud (VPC).

#### Get-Item and Get-ChildItem

#### HypervisorConnectionObjectNotFound

The hypervisor connection object specified in the path could not be found.

#### HostingUnitObjectNotFound

The specified hosting unit object could not be found.

#### HypervisorInMaintenanceMode

The hypervisor for the specified connection is currently in maintenance mode and cannot be accessed.

#### FailureToRetrieveConnectionPassword

The password for the HypervisorConnection cannot be retrieved or decrypted.

#### ExceptionThrown

An unexpected error occurred.

#### DatabaseError

There was a problem communicating with the database.

#### CommunicationError

There was a problem communicating with the remote service.

### Remove-Item

#### HostingUnitObjectToDeleteDoesNotExist

The specified hosting unit object does not exist.

#### HypervisorConnectionObjectToDeleteDoesNotExist

The specified hypervisor connection object does not exist.

#### InputPathInvalid

The specified path is invalid because it is not in one of the following formats:

XDHyp:\Connections\<Name>

XDHyp:\Connections\{Guid}

XDHyp:\HostingUnits\<Name>

XDHyp:\HostingUnits\{Guid}

#### HypervisorConnectionObjectToDeleteIsInUse

The specified hypervisor connection object cannot be deleted because it is being used by one or more hosting units.

#### ExceptionThrown

An unexpected error occurred.

#### DatabaseError

There was a problem communicating with the database.

#### CommunicationError

There was a problem communicating with the remote service.

#### ObjectCannotBeRemoved

The specified object cannot be removed.

## Rename-Item

### InputPathInvalid

The specified path is invalid because it is not in one of the following formats:

XDHyp:\Connections\XDHyp:\Connections\{Guid}  
XDHyp:\HostingUnits\XDHyp:\HostingUnits\{Guid}

### HostingUnitDuplicateNameExists

A hosting unit object with the same name already exists. Hosting unit names must be unique.

### HypervisorConnectionDuplicateNameExists

A hypervisor connection object with the same name already exists. Hypervisor connection names must be unique.

### InputNameInvalid

The new name specified for the hosting unit or hypervisor connection is invalid because it contains one or more of the following characters: \;#.\*?=<>|[](){}.

### ExceptionThrown

An unexpected error occurred.

### DatabaseError

There was a problem communicating with the database.

### CommunicationError

There was a problem communicating with the remote service.

## Set-Item

### InputPathInvalid

The specified path is invalid because it is not in one of the following formats:

XDHyp:\Connections\XDHyp:\Connections\{Guid}  
XDHyp:\HostingUnits\XDHyp:\HostingUnits\{Guid}

### HostingUnitObjectToUpdateDoesNotExist

The specified hosting unit object does not exist.

#### HostingUnitNetworkPathInvalid

The new network path specified for the hosting unit is invalid.  
Either the network path does not exist or it is not relative to the root path of the hosting unit.

#### HypervisorConnectionObjectToUpdateDoesNotExist

The specified hypervisor connection object does not exist.

#### ExceptionThrown

An unexpected error occurred.

#### DatabaseError

There was a problem communicating with the database.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation.  
Communication with the database failed for various reasons.

#### CommunicationError

There was a problem communicating with the remote service.

# about\_Hyp\_Filtering

Jan 04, 2017

## TOPIC

XenDesktop - Advanced Dataset Filtering

## SHORT DESCRIPTION

Describes the common filtering options for XenDesktop cmdlets.

## LONG DESCRIPTION

Some cmdlets operate on large quantities of data and, to reduce the overhead of sending all of that data over the network, many of the Get- cmdlets support server-side filtering of the results.

The conventional way of filtering results in PowerShell is to pipeline them into Where-Object, Select-Object, and Sort-Object, for example:

```
Get-<Noun> | Where { $_.Size = 'Small' } | Sort 'Date' | Select -First 10
```

However, for most XenDesktop cmdlets the data is stored remotely and it would be slow and inefficient to retrieve large amounts of data over the network and then discard most of it. Instead, many of the Get- cmdlets provide filtering parameters that allow results to be processed on the server, returning only the required results.

You can filter results by most object properties using parameters derived from the property name. You can also sort results or limit them to a specified number of records:

```
Get-<Noun> -Size 'Small' -SortBy 'Date' -MaxRecordCount 10
```

You can express more complex filter conditions using a syntax and set of operators very similar to those used by PowerShell expressions.

Those cmdlets that support filtering have the following common parameters:

-MaxRecordCount <int>

Specifies the maximum number of results to return.  
For example, to return only the first nine results use:

```
Get-<Noun> -MaxRecordCount 9
```

If not specified, only the first 250 records are returned, and if more are available, a warning is produced:



WARNING: Only first 250 records returned. Use -MaxRecordCount to retrieve more.

You can suppress this warning by using -WarningAction or by specifying a value for -MaxRecordCount.

To retrieve all records, specify a large number for -MaxRecordCount. As the value is an integer, you can use the following:

```
Get-<Noun> -MaxRecordCount [int]::MaxValue
```

-ReturnTotalRecordCount [<SwitchParameter>]

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. For example:

```
Get-<Noun> -MaxRecordCount 9 -ReturnTotalRecordCount
```

....

```
Get-<Noun> : Returned 9 of 10 items
```

```
At line:1 char:18
```

```
+ Get-<Noun> <<<< -MaxRecordCount 9 -ReturnTotalRecordCount
```

```
+ CategoryInfo          : OperationStopped: (:) [Get-<Noun>], PartialDataException
```

```
+ FullyQualifiedErrorId : PartialData,Citrix.<SDKName>.SDK.Get<Noun>
```

The count can be accessed using the TotalAvailableResultCount property:

```
$count = $error[0].TotalAvailableResultCount
```

-Skip <int>

Skips the specified number of records before returning results.

Also reduces the count returned by -ReturnTotalRecordCount.

-SortBy <string>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order, respectively. Ascending order is assumed if no prefix is present.

Sorting occurs before `-MaxRecordCount` and `-Skip` parameters are applied. For example, to sort by Name and then by Count (largest first) use:

```
-SortBy 'Name,-Count'
```

By default, sorting by an enumeration property uses the numeric value of the elements. You can specify a different sort order by qualifying the name with an ordered list of elements or their numeric values, or `<null>` to indicate the placement of null values.

Elements not mentioned are placed at the end in their numeric order. For example, to sort by two different enums and then by the object id:

```
-SortBy 'MyState(StateC,<null>,StateA,StateB),Another(0,3,2,1),Id'
```

`-Filter <String>`

This parameter lets you specify advanced filter expressions, and supports combination of conditions with `-and` and `-or`, and grouping with braces. For example:

```
Get-<Noun> -Filter 'Name -like "High*" -or (Priority -eq 1 -and Severity -ge 2)'
```

The syntax is close enough to PowerShell syntax that you can use script blocks in most cases. This can be easier to read as it reduces quoting:

```
Get-<Noun> -Filter { Count -ne $null }
```

The full `-Filter` syntax is provided below.

## EXAMPLES

Filtering by strings performs a case-insensitive wildcard match.

Separate parameters are combined with an implicit `-and` operator.

Normal PowerShell quoting rules apply, so you can use single or double quotes, and omit the quotes altogether for many strings. The order of parameters does not make any difference. The following are equivalent:

```
Get-<Noun> -Company Citrix -Product Xen*
Get-<Noun> -Company "citrix" -Product '[X]EN*'
Get-<Noun> -Product "Xen*" -Company "CITRIX"
Get-<Noun> -Filter { Company -eq 'Citrix' -and Product -like 'Xen*' }
```

See [about\\_Quoting\\_Rules](#) and [about\\_Wildcards](#) for details about PowerShell handling of quotes and wildcards.

To avoid wildcard matching or include quote characters, you can escape the wildcards using the normal PowerShell escape mechanisms (see [about\\_Escape\\_Characters](#)), or switch to a filter expression and the `-eq` operator:

```
Get-<Noun> -Company "Abc[*]"           # Matches Abc*
Get-<Noun> -Company "Abc`*"           # Matches Abc*
Get-<Noun> -Filter { Company -eq "Abc*" } # Matches Abc*
Get-<Noun> -Filter { Company -eq "A`"B`C" } # Matches A"B'C
```

Simple filtering by numbers, booleans, and TimeSpans perform direct equality comparisons, although if the value is nullable you can also search for null values. Here are some examples:

```
Get-<Noun> -Uid 123
Get-<Noun> -Enabled $true
Get-<Noun> -Duration 1:30:40
Get-<Noun> -NullableProperty $null
```

More comparisons are possible using advanced filtering with `-Filter`:

```
Get-<Noun> -Filter 'Capacity -ge 10gb'
Get-<Noun> -Filter 'Age -ge 20 -and Age -lt 40'
Get-<Noun> -Filter 'VolumeLevel -like "[123]"'
Get-<Noun> -Filter 'Enabled -ne $false'
Get-<Noun> -Filter 'NullableProperty -ne $null'
```

You can check boolean values without an explicit comparison operator, and you can also combine them with `-not`:

```
Get-<Noun> -Filter 'Enabled' # Equivalent to 'Enabled -eq $true'
Get-<Noun> -Filter '-not Enabled' # Equivalent to 'Enabled -eq $false'
```

See [about\\_Comparison\\_Operators](#) for an explanation of the operators, but note that only a subset of PowerShell operators are supported (`-eq`, `-ne`, `-gt`, `-ge`, `-lt`, `-le`, `-like`, `-notlike`, `-in`, `-notin`, `-contains`, `-notcontains`).

Enumeration values can either be specified using typed values or the string name of the enumeration value:

```
Get-<Noun> -Shape [Shapes]::Square
Get-<Noun> -Shape Circle
```

With filter expressions, typed values can be specified with simple variables or quoted strings. They also support enumerations with wildcards:

```
$s = [Shapes]::Square
Get-<Noun> -Filter { Shape -eq $s -or Shape -eq "Circle" }
Get-<Noun> -Filter { Shape -like 'C*' }
```

By their nature, floating point values, DateTime values, and TimeSpan values are best suited to relative comparisons rather than just equality. DateTime strings are converted using the locale and time zone of the user device, but you can use ISO8601 format strings (YYYY-MM-DDThh:mm:ss.sTZD) to avoid ambiguity. You can also use standard PowerShell syntax to create these values:

```
Get-<Noun> -Filter { StartTime -ge "2010-08-23T12:30:00.0Z" }
$d = [DateTime]"2010-08-23T12:30:00.0Z"
Get-<Noun> -Filter { StartTime -ge $d }
$d = (Get-Date).AddDays(-1)
Get-<Noun> -Filter { StartTime -ge $d }
```

Relative times are quite common and, when using filter expressions, you can also specify DateTime values using a relative format:

```
Get-<Noun> -Filter { StartTime -ge '-2' } # Two days ago
Get-<Noun> -Filter { StartTime -ge '-1:30' } # Hour and a half ago
Get-<Noun> -Filter { StartTime -ge '-0:0:30' } # 30 seconds ago
```

## ARRAY PROPERTIES

When filtering against list or array properties, simple parameters perform a case-insensitive wildcard match against each of the members. With filter expressions, you can use the -contains and -notcontains operators. Unlike PowerShell, these perform wildcard matching on strings.

Note that for array properties the naming convention is for the returned property to be plural, but the parameter used to search for any match is singular. The following are equivalent (assuming Users is an array property):

```
Get-<Noun> -User Fred*
Get-<Noun> -Filter { User -like "Fred*" }
Get-<Noun> -Filter { Users -contains "Fred*" }
```

You can also use the singular form with -Filter to search using other operators:

```
# Match if any user in the list is called "Frederick"
Get-<Noun> -Filter { User -eq "Frederick" }
# Match if any user in the list has a name alphabetically below 'F'
Get-<Noun> -Filter { User -lt 'F' }
```

## COMPLEX EXPRESSIONS

When matching against multiple values, you can use a sequence of comparisons joined with -or operators, or you can use -in and -notin:

```
Get-<Noun> -Filter { Shape -eq 'Circle' -or Shape -eq 'Square' }
$shapes = 'Circle','Square'
Get-<Noun> -Filter { Shape -in $shapes }
$sides = 1..4
Get-<Noun> -Filter { Sides -notin $sides }
```

Braces can be used to group complex expressions, and override the default left-to-right evaluation of -and and -or. You can also use -not to invert the sense of any sub-expression:

```
Get-<Noun> -Filter { Size -gt 4 -or (Color -eq 'Blue' -and Shape -eq 'Circle') }
Get-<Noun> -Filter { Sides -lt 5 -and -not (Color -eq 'Blue' -and Shape -eq 'Circle') }
```

## PAGING

The simplest way to page through data is to use the -Skip and -MaxRecordCount parameters. So, to read the first three pages of data with 10 records per page, use:

```
Get-<Noun> -Skip 0 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 10 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 20 -MaxRecordCount 10 <other filtering criteria>
```

You must include the same filtering criteria on each call, and ensure that the data is sorted consistently.

The above approach is often acceptable, but as each call performs an independent query, data changes can result in records being skipped or appearing twice. One approach to improve this is to sort by a unique id field and then start the search for the next page at the unique id after the last unique id of the previous page. For example:

```
# Get the first page
Get-<Noun> -MaxRecordCount 10 -SortBy SerialNumber

SerialNumber ...
----- ---
A120004
A120007
... 7 other records ...
A120900

# Get the next page
Get-<Noun> -MaxRecordCount 10 -Filter { FirstName -gt 'A120900' }

SerialNumber ...
```

----- ---  
A120901  
B220000  
...

## FILTER SYNTAX DEFINITION

<Filter> ::= <ScriptBlock> | <ComponentList>

<ScriptBlock> ::= "{" <ComponentList> "}"

<ComponentList> ::= <Component> <AndOrOperator> <ComponentList> |

<Component>

<Component> ::= <NotOperator> <Factor> |

<Factor>

<Factor> ::= "(" <ComponentList> ")" |

<PropertyName> <ComparisonOperator> <Value> |  
<PropertyName>

<AndOrOperator> ::= "-and" | "-or"

<NotOperator> ::= "-not" | "!"

<ComparisonOperator>

::= "-eq" | "-ne" | "-le" | "-ge" | "-lt" | "-gt" |  
"-like" | "-notlike" | "-contains" | "-notcontains" |  
"-in" | "-notin"

<PropertyName> ::= <simple name of property>

<Value> ::= <string literal> | <numeric literal> |

<scalar variable> | <array variable> |  
"\$null" | "\$true" | "\$false"

Numeric literals support decimal and hexadecimal literals, with optional multiplier suffixes (kb, mb, gb, tb, pb).

Dates and times can be specified as string literals. The current culture determines what formats are accepted. To avoid any ambiguity, use strings formatted to the ISO8601 standard. If not specified, the current time zone is used.

Relative date-time string literals are also supported, using a minus sign followed by a TimeSpan. For example, "-1:30" means 1 hour and 30 minutes ago.

# Add-HypHostingUnitMetadata

Jan 04, 2017

Adds metadata on the given HostingUnit.

## Syntax

```
Add-HypHostingUnitMetadata [-HostingUnitUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-HypHostingUnitMetadata [-HostingUnitUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-HypHostingUnitMetadata [-HostingUnitName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-HypHostingUnitMetadata [-HostingUnitName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-HypHostingUnitMetadata [-InputObject] <HostingUnit[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-HypHostingUnitMetadata [-InputObject] <HostingUnit[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given HostingUnit objects. This cmdlet will not overwrite existing metadata on an object - use the Set-HypHostingUnitMetadata cmdlet instead.

## Related topics

[Set-HypHostingUnitMetadata](#)

[Remove-HypHostingUnitMetadata](#)

## Parameters

**-HostingUnitUid**<Guid>

Id of the HostingUnit

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-HostingUnitName**<String>

Name of the HostingUnit



|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<HostingUnit[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the HostingUnit specified. The property cannot contain any of the following characters \/:;#.\*?=<> |[]()"

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |       |
|------------------------|-------|
| Default Value          |       |
| Accept Pipeline Input? | false |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

Return Values

Citrix.Host.Sdk.Metadata

Add-HypHostingUnitMetadata returns an array of objects containing the new definition of the metadata.

\n Property <string>

\n Specifies the name of the property.

\n Value <string>

\n Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DuplicateObject

One of the specified metadata already exists.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Add-HypHostingUnitMetadata -HostingUnitUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Property | Value |
|----------|-------|
| -----    | ----- |
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the HostingUnit with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Add-HypHostingUnitNetwork

Jan 04, 2017

Makes additional hypervisor networks available for use in a HostingUnit.

## Syntax

```
Add-HypHostingUnitNetwork [-LiteralPath] <String> [-NetworkPath] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this command to extend the set of hypervisor networks that are made available through the HostingUnit to the Citrix Machine Creation Service. When new machines are created, their virtual NICs can be associated only with networks that are in this set. This command cannot be used if the connection for the hosting unit is in maintenance mode.

## Related topics

[New-Item](#)

[Add-HypMetadata](#)

[remove-HypHostingUnitNetwork](#)

## Parameters

### **-LiteralPath**<String>

Specifies the path to the hosting unit to which the network will be added. The path must be in one of the following formats: <drive>:\HostingUnits\  
<HostingUnitName> or <drive>:\HostingUnits\{<HostingUnit Uid>}

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-NetworkPath**<String>

Specifies the path to the network that will be added. The path must be in one of the following formats: <drive>:\Connections\  
<ConnectionName>\MyNetwork.network or <drive>:\Connections\{<Connection Uid>\MyNetwork.network or <drive>:\HostingUnits\  
<HostingUnitName>\MyNetwork.network or <drive>:\HostingUnits\{<hostingUnit Uid>\MyNetwork.network

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. This can be a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | LocalHost. When a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Input Type

System.string You can pipe a string that contains a path to Add-HypHostingUnitNetwork (NetworkPath parameter).

### Return Values

Citrix.Host.Sdk.HostingUnit

Add-HypHostingUnitNetwork returns an object containing the new definition of the hosting unit.

HostingUnitUid <Guid>

Specifies the unique identifier for the hosting unit.

HostingUnitName <string>

Specifies the name of the hosting unit.

HypervisorConnection <Citrix.Host.Sdk.HypervisorConnection>

Specifies the connection that the hosting unit uses to access a hypervisor.

RootId <string>

Identifies, to the hypervisor, the root of the hosting unit.

RootPath <string>

The hosting unit provider path that represents the root of the hosting unit.

Storage <Citrix.Host.Sdk.Storage[]>

The list of storage items that the hosting unit can use.

PersonalDiskStorage <Citrix.XDPowerShell.Storage[]>

The list of storage items that the hosting unit can use for storing personal data.

VMTaggingEnabled <Boolean>

Specifies whether or not the metadata in the hypervisor can be used to store information about the XenDesktop Machine Creation Service.

NetworkId <string>

The hypervisor's internal identifier that represents the default network specified for the hosting unit.

NetworkPath <string>

The hosting unit provider path to the default network specified for the hosting unit.

Metadata <Citrix.Host.Sdk.Metadata[]>

A list of key value pairs that can store additional information about the hosting unit.

PermittedNetworks <Citrix.Host.Sdk.Network[]>

A full list of the hypervisor networks that are exposed for use in the hosting unit.

### Notes

The network path must be valid for the hosting unit. The rules that are applied are as follows: XenServer (HypervisorConnection Type = XenServer)

NA

VMWare vSphere/ESX (HypervisorConnection Type = vCenter)

The network path must be directly contained in the root path item of the hosting unit.

Microsoft SCVMM/Hyper-v (HypervisorConnection Type = SCVMM)

NA

In the case of failure, the following errors can result.

Error Codes

-----

HostingUnitsPathInvalid

The path provided is not to an item in a subdirectory of a hosting unit item.

HostingUnitNetworkPathInvalid

The specified path is invalid.

HostingUnitNetworkPathInvalid

The network path cannot be found or is invalid. See notes above about validity.

HostingUnitNetworkDuplicateObjectExists

The specified network path is already part of the hosting unit.

HypervisorInMaintenanceMode

The hypervisor for the connection is in maintenance mode.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation. Communication with the database failed for various reasons.

CommunicationError

An error occurred while communicating with the service.

PermissionDenied

The user does not have administrative rights to perform this operation.

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used, or examine the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Add-HypHostingUnitNetwork -LiteralPath XDHyp:\HostingUnits\MyHostingUnit -NetworkPath 'XDHyp:\HostingUnits\MyHostingUnits\newNetwork.network'
```

```
HostingUnitUid      : bcd3d649-86d1-4aa8-8ec2-d322b6a2c457
HostingUnitName     : MyHostingUnit
```

```

HypervisorConnection : MyConnection
RootPath             : /
RootId               :
NetworkPath          : /Network 0.network
NetworkId            : ab47080b-ca15-771a-c8dc-6ad9650158f1
Storage              : {/Local storage.storage}
PersonalvDiskStorage : {}
VMTaggingEnabled    : True
Metadata             : {}
PermittedNetworks   : {/Network 0.network, /newNetwork.network}

```

The command adds a new network called "newNetwork.network" to the hosting unit called "MyHostingUnit".

----- **EXAMPLE 2** -----

```

XDHyp:\HostingUnits\MyHostingUnit>Add-HypHostingUnitNetwork -LiteralPath . -NetworkPath 'XDHyp:\HostingUnits\MyHostingUnits\newNetwork.network'

```

```

HostingUnitUid       : bcd3d649-86d1-4aa8-8ec2-d322b6a2c457
HostingUnitName      : MyHostingUnit
HypervisorConnection : MyConnection
RootPath             : /
RootId               :
NetworkPath          : /Network 0.network
NetworkId            : ab47080b-ca15-771a-c8dc-6ad9650158f1
Storage              : {/Local storage.storage}
PersonalvDiskStorage : {}
VMTaggingEnabled    : True
Metadata             : {}
PermittedNetworks   : {/Network 0.network, /newNetwork.network}

```

The command adds a new network called "newNetwork.network" to the current directory. The dot (.) represents the current location (not its contents).

----- **EXAMPLE 3** -----

```

XDHyp:\HostingUnits\MyHostingUnit>dir *.network | Add-HypHostingUnitNetwork -LiteralPath XDHyp:\HostingUnits\MyHostingUnit

```

The command adds all of the networks that are available in the hosting unit to the specified hosting unit.

----- **EXAMPLE 4** -----

```

c:\PS>Add-HypHostingUnitNetwork -LiteralPath XDHyp:\HostingUnits\MyHostingUnit -NetworkPath 'XDHyp:\HostingUnits\MyHostingUnits\newNetwork.network'

```

```

HostingUnitUid       : bcd3d649-86d1-4aa8-8ec2-d322b6a2c457
HostingUnitName      : MyHostingUnit
HypervisorConnection : MyConnection
RootPath             : /
RootId               :
NetworkPath          : /Network 0.network
NetworkId            : ab47080b-ca15-771a-c8dc-6ad9650158f1
Storage              : {/Local storage.storage}
PersonalvDiskStorage : {}
VMTaggingEnabled    : True
Metadata             : {}
PermittedNetworks   : {/Network 0.network, /newNetwork.network}

```

The command adds a new network location called "newNetwork.network" to the hosting unit called "MyHostingUnit".

# Add-HypHostingUnitStorage

Jan 04, 2017

Adds storage locations to a hosting unit.

## Syntax

```
Add-HypHostingUnitStorage [-LiteralPath] <String> [-StoragePath] <String> [-StorageType <StorageType>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this command to add storage locations for storing the hard disks required by the virtual machines created by the Citrix Machine Creation Service. You cannot use this command if the connection for the hosting unit is in maintenance mode.

## Related topics

[New-Item](#)

[Add-HypMetadata](#)

[remove-HypHostingUnitStorage](#)

## Parameters

### **-LiteralPath**<String>

Specifies the path to the hosting unit to which storage will be added. The path must be in one of the following formats: <drive>:\HostingUnits\<HostingUnitName> or <drive>:\HostingUnits\{<HostingUnit Uid>}

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-StoragePath**<String>

Specifies the path to the storage that will be added. The path must be in one of the following formats: <drive>:\Connections\<ConnectionName>\MyStorage.storage or <drive>:\Connections\{<Connection Uid>}\MyStorage.storage or <drive>:\HostingUnits\<HostingUnitName>\MyStorage.storage or <drive>:\HostingUnits\{<HostingUnit Uid>}\MyStorage.storage

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

### **-StorageType**<StorageType>

Specifies the type of storage in StoragePath. Supported storage types are: OSStorage PersonalDiskStorage

|                        |           |
|------------------------|-----------|
| Required?              | false     |
| Default Value          | OSStorage |
| Accept Pipeline Input? | false     |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. This can be a host name or an IP address.



|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | LocalHost. When a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

#### Input Type

Systemstring You can pipe a string that contains a path to Add-HypHostingUnitStorage (StoragePath parameter).

#### Return Values

Citrix.Host.Sdk.HostingUnit

Add-HypHostingUnitStorage returns an object containing the new definition of the hosting unit.

HostingUnitId <Guid>

Specifies the unique identifier for the hosting unit.

HostingUnitName <string>

Specifies the name of the hosting unit.

HypervisorConnection <Citrix.Host.Sdk.HypervisorConnection>

Specifies the connection that the hosting unit uses to access a hypervisor.

RootId <string>

Identifies, to the hypervisor, the root of the hosting unit.

RootPath <string>

The hosting unit provider path that represents the root of the hosting unit.

Storage <Citrix.Host.Sdk.Storage[]>

The list of storage items that the hosting unit can use.

PersonalDiskStorage <Citrix.XDPowerShell.Storage[]>

The list of storage items that the hosting unit can use for storing personal data.

VMTaggingEnabled <Boolean>

Specifies whether or not the metadata in the hypervisor can be used to store information about the XenDesktop Machine Creation Service.

NetworkId <string>

The hypervisor's internal identifier that represents the network specified for the hosting unit.

NetworkPath <string>

The hosting unit provider path to the network specified for the hosting unit.

Metadata <Citrix.Host.Sdk.Metadata[]>

A list of key value pairs that can store additional information about the hosting unit.

#### Notes

The storage path must be valid for the hosting unit. The rules that are applied are as follows: XenServer (HypervisorConnection Type = XenServer)

NA

VMWare vSphere/ESX (HypervisorConnection Type = vCenter)

The storage path must be directly contained in the root path item of the hosting unit.

Microsoft SCVMM/Hyper-v (HypervisorConnection Type = SCVMM)

Only one storage entry for these connection types is valid, and it must reference an SMB share. Additionally, if a Hyper-V failover cluster is used the SMB share must be the top-level mount point of the cluster shared volume on one of the servers in the cluster (i.e. C:\ClusterStorage).

In the case of failure, the following errors can result.

#### Error Codes

-----

HostingUnitsPathInvalid

The path provided is not to an item in a subdirectory of a hosting unit item.

HostingUnitStoragePathInvalid

The specified path is invalid.

HostingUnitStoragePathInvalid

The storage path cannot be found or is invalid. See notes above about validity.

HostingUnitStorageDuplicateObjectExists

The specified storage path is already part of the hosting unit.

HypervisorInMaintenanceMode

The hypervisor for the connection is in maintenance mode.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation. Communication with the database failed for various reasons.

CommunicationError

An error occurred while communicating with the service.

PermissionDenied

The user does not have administrative rights to perform this operation.

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used, or examine the XenDesktop logs.

Examples

----- EXAMPLE 1 -----

```
c:\PS>Add-HypHostingUnitStorage -LiteralPath XDHyp:\HostingUnits\MyHostingUnit -StoragePath 'XDHyp:\HostingUnits\MyHostingUnits\newStorage.storage'
```

```
HostingUnitUid      : bcd3d649-86d1-4aa8-8ec2-d322b6a2c457
HostingUnitName     : MyHostingUnit
HypervisorConnection : MyConnection
RootPath            : /
RootId              :
NetworkPath         : /Network 0.network
NetworkId           : ab47080b-ca15-771a-c8dc-6ad9650158f1
Storage             : {/Local storage.storage, /newStorage.storage}
PersonalvDiskStorage : {/newStorage.storage}
VMTaggingEnabled    : True
Metadata            : {}
```

The command adds a new storage location called "newStorage.storage" to the hosting unit called "MyHostingUnit".

----- EXAMPLE 2 -----

```
XDHyp:\HostingUnits\MyHostingUnit>Add-HypHostingUnitStorage -LiteralPath . -StoragePath 'XDHyp:\HostingUnits\MyHostingUnits\newStorage.storage' -StorageType OSSStorage
```

```
HostingUnitUid      : bcd3d649-86d1-4aa8-8ec2-d322b6a2c457
HostingUnitName     : MyHostingUnit
HypervisorConnection : MyConnection
RootPath            : /
RootId              :
NetworkPath         : /Network 0.network
NetworkId           : ab47080b-ca15-771a-c8dc-6ad9650158f1
Storage             : {/Local storage.storage, /newStorage.storage}
PersonalvDiskStorage : {/Local storage.storage}
VMTaggingEnabled    : True
Metadata            : {}
```

The command adds a new storage location called "newStorage.storage" to the current directory. The dot (.) represents the current location (not its contents).

----- EXAMPLE 3 -----

```
XDHyp:\HostingUnits\MyHostingUnit>dir *.storage | Add-HypHostingUnitStorage -LiteralPath XDHyp:\HostingUnits\MyHostingUnit
```

The command adds all of the storage that is available in the hosting unit to the specified hosting unit.

----- EXAMPLE 4 -----

```
c:\PS>Add-HypHostingUnitStorage -LiteralPath XDHyp:\HostingUnits\MyHostingUnit -StoragePath 'XDHyp:\HostingUnits\MyHostingUnits\newStorage.storage' -StorageType PersonalvDiskStorage
```

```
HostingUnitUid    : bcd3d649-86d1-4aa8-8ec2-d322b6a2c457
HostingUnitName   : MyHostingUnit
HypervisorConnection : MyConnection
RootPath          : /
RootId            :
NetworkPath       : /Network 0.network
NetworkId         : ab47080b-ca15-771a-c8dc-6ad9650158f1
Storage           : {/Local storage.storage}
PersonalvDiskStorage : {/Local storage.storage, /newStorage.storage}
VMTaggingEnabled  : True
Metadata          : {}
```

The command adds a new storage location called "newStorage.storage" to the hosting unit called "MyHostingUnit".

# Add-HypHypervisorConnectionAddress

Jan 04, 2017

Add a connection address to a hypervisor connection.

## Syntax

```
Add-HypHypervisorConnectionAddress [-LiteralPath] <String> [-HypervisorAddress] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this command to add addresses by which a hypervisor can be contacted. All addresses added to a single hypervisor connection are assumed to be equivalent (i.e. they all result in the ability to communicate with the same hypervisors). The hypervisor that the addresses reference is stored at the point of creation of the hypervisor connection. Once this is done, to be valid, all addresses must resolve to this hypervisor.

The addresses required are; XenServer - The address of the XenServer machines (must all reference the same XenServer pool). VMWare vSphere/ESX - The address of a vCenter Server. Microsoft SCVMM/Hyper-V - the address of an SCVMM server.

## Related topics

[Remove-HypHypervisorConnectionAddress](#)

[Get-HypXenServerAddress](#)

## Parameters

**-LiteralPath**<String>

Specifies the path within a Host Service provider to the hypervisor connection item to which to add the address. The path specified must be in one of the following formats; <drive>:\Connections\<HypervisorConnectionName> or <drive>:\Connections\{HypervisorConnection Uid>}

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-HypervisorAddress**<String>

Specifies the address to be used. The address will be validated and the hypervisor must be contactable at the address supplied.

XenServer (ConnectionType = XenServer) The address being added must reference the same XenServer pool referenced by any existing addresses for the same connection. VMware vSphere\ESX (ConnectionType = vCenter) The address being added must be to the same VMware vCenter as the existing addresses. SCVMM\Hyper-v (ConnectionType = SCVMM) The address being added must be to the same SCVMM server as the existing addresses. Custom Connection Types ?Validation here I assume none need to check?

#PUBS NOTE: ASSUME THIS TO BE UPDATED?#

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                |
|------------------------|----------------|
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

**Return Values**

Citrix.Host.Sdk.HypervisorConnection

Add-HypHypervisorConnectionAddress returns an object containing the new definition of the hypervisor connection.

HypervisorConnectionUid <Guid>

Specifies the unique identifier for the hypervisor connection.

HypervisorConnectionName <string>

Specifies the name of the hypervisor connection.

ConnectionType <Citrix.XDInterServiceTypes.ConnectionType>

Specifies the connection type of the connection.

XenServer - A Citrix XenServer hypervisor

SCVMM - A Microsoft SCVMM/Hyper-V hypervisor

vCenter - A VMWare vSphere/ESX hypervisor

Custom - A 3rd party hypervisor

HypervisorAddress <string[]>

A list of addresses that can be used to communicate with the hypervisor.

UserName <string>

The user name that is used when connecting to the hypervisor.

Persistent <Boolean>

Indicates whether the connection is stored in the database or is a temporary connection only with the same lifetime as the current Powershell session.

PlugInId <string>

The Citrix identifier for the Citrix Machine Management plug-in.

Revision <Guid>

Identifier for the current version of the hypervisor connection. This value changes every time a property of the hypervisor connection is updated.

SupportsPvsVMs <Boolean>

Indicates whether or not the connection can be used as part of a hosting unit and therefore used by the Citrix XenDesktop Machine Creation Service to create virtual machines. Only the built-in supported hypervisor connection types can be used for this (i.e. XenServer, SCVMM and vCenter).

Metadata <Citrix.Host.Sdk.Metadata[]>

A list of key value pairs that can store additional information relating to the hosting unit.

Notes

The address format must be valid for the hypervisor connection. The rules that are applied are as below: XenServer (HypervisorConnection Type = XenServer)

http:\\<IP Address>

or http:\\<server Name>

or https:\\<server Name>

Note: To use multiple addresses to the same XenServer pool to provide failover functionality, all XenServers in the pool must have a shared block storage device. If the use of https connections and failover is required, the certificates on the servers must be trusted by all of the controllers (typically this means having a root certificate installed).

VMWare vSphere/ESX (HypervisorConnection Type = vCenter)

http:\\<IP Address>\\<sdk path>

http:\\<server Name>\\<sdk path>

or https:\\<Server Name>\\<sdk path>

Notes:

\* Http requires manual vCenter configuration; https is the only supported default format.

\* Default sdk path is 'sdk'

Microsoft SCVMM/Hyper-v (HypervisorConnection Type = SCVMM)

IP Address

server Name (short or FQDN)

Note: If XenServers are moved to new XenServer pools after being added to a hypervisor connection, unpredictable results can occur. Once a XenServer is assigned to a hypervisor connection, it must not be moved to a new XenServer pool.

In the case of failure the following errors can result.

Error Codes

-----

InputConnectionsPathInvalid

The path provided is not to an item in a sub directory of a hosting unit item.

HypervisorConnectionAddressForeignKeyObjectDoesNotExist

The hypervisor connection to which the address is to be added could not be located.

ConnectionAddressInvalid

The address could not be used to contact a hypervisor or the validation rules have not been met.

HypervisorConnectionAddressDuplicateObjectExists

The address specified is already part of the hypervisor connection.

HypervisorInMaintenanceMode

The hypervisor for the connection is in maintenance mode.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

CommunicationError

An error occurred while communicating with the service.

PermissionDenied

The user does not have administrative rights to perform this operation.

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Add-HypHypervisorConnectionAddress -LiteralPath XDHyp:\Connections\MyConnection -HypervisorAddress http:\\myserver.com
```

```
PSPath           : Citrix.HostingUnitService.Admin.V1.0\Citrix.Hypervisor::XDHyp:\Connections\MyConnection
PSParentPath     : Citrix.HostingUnitService.Admin.V1.0\Citrix.Hypervisor::XDHyp:\Connections
PSChildName      : MyConnection
PSDrive          : XDHyp
PSProvider       : Citrix.HostingUnitService.Admin.V1.0\Citrix.Hypervisor
PSIsContainer    : True
HypervisorConnectionUid : 85581f42-c5da-4976-970c-ebc3448ea1e3
HypervisorConnectionName : MyConnection
ConnectionType   : XenServer
HypervisorAddress : {http:\\myserver2.com,http:\\myserver.com}
UserName         : root
Persistent       : False
PluginId        : XenFactory
SupportsPvsVMs  : True
Revision         : 4c95c857-c54d-4f92-abef-0cce32c02502
Metadata        :
```

Add the address 'http:\\myserver.com' to the hypervisor connection called "MyConnection".



# Add-HypHypervisorConnectionMetadata

Jan 04, 2017

Adds metadata on the given HypervisorConnection.

## Syntax

```
Add-HypHypervisorConnectionMetadata [-HypervisorConnectionUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-HypHypervisorConnectionMetadata [-HypervisorConnectionUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-HypHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-HypHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-HypHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-HypHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given HypervisorConnection objects. This cmdlet will not overwrite existing metadata on an object - use the Set-HypHypervisorConnectionMetadata cmdlet instead.

## Related topics

[Set-HypHypervisorConnectionMetadata](#)

[Remove-HypHypervisorConnectionMetadata](#)

## Parameters

**-HypervisorConnectionUid**<Guid>

Id of the HypervisorConnection

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-HypervisorConnectionName**<String>

Name of the HypervisorConnection

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-InputObject**<HypervisorConnection[]>

Objects to which the metadata is to be added.

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the HypervisorConnection specified. The property cannot contain any of the following characters \/:;#.\*?=<>|[]0"

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Direct or typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|               |       |
|---------------|-------|
| Required?     | false |
| Default Value |       |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

#### Return Values

Citrix.Host.Sdk.Metadata

Add-HypervisorConnectionMetadata returns an array of objects containing the new definition of the metadata.

\n Property <string>

\n Specifies the name of the property.

\n Value <string>

\n Specifies the value for the property.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DuplicateObject

One of the specified metadata already exists.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Add-HypHypervisorConnectionMetadata -HypervisorConnectionUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

| Property | Value |
|----------|-------|
| property | value |

Add metadata with a name of 'property' and a value of 'value' to the HypervisorConnection with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Add-HypHypervisorConnectionScope

Jan 04, 2017

Add the specified HypervisorConnection(s) to the given scope(s).

## Syntax

```
Add-HypHypervisorConnectionScope [-Scope] <String[]> -InputObject <HypervisorConnection[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-HypHypervisorConnectionScope [-Scope] <String[]> -HypervisorConnectionUid <Guid[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-HypHypervisorConnectionScope [-Scope] <String[]> -HypervisorConnectionName <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The AddHypHypervisorConnectionScope cmdlet is used to associate one or more HypervisorConnection objects with given scope(s).

There are multiple parameter sets for this cmdlet, allowing you to identify the HypervisorConnection objects in different ways:

- HypervisorConnection objects can be piped in or specified by the InputObject parameter
- The HypervisorConnectionUid parameter specifies objects by HypervisorConnectionUid
- The HypervisorConnectionName parameter specifies objects by HypervisorConnectionName (supports wildcards)

To add a HypervisorConnection to a scope you need permission to change the scopes of the HypervisorConnection and permission to add objects to all of the scopes you have specified.

If the HypervisorConnection is already in a scope, that scope will be silently ignored.

## Related topics

[Remove-HypHypervisorConnectionScope](#)

[Get-HypScopedObject](#)

## Parameters

**-Scope**<String[]>

Specifies the scopes to add the objects to.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-InputObject**<HypervisorConnection[]>

Specifies the HypervisorConnection objects to be added.

|           |      |
|-----------|------|
| Required? | true |
|-----------|------|

|                        |                                |
|------------------------|--------------------------------|
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-HypervisorConnectionUid**<Guid[]>

Specifies the HypervisorConnection objects to be added by HypervisorConnectionUid.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-HypervisorConnectionName**<String[]>

Specifies the HypervisorConnection objects to be added by HypervisorConnectionName.

|                        |                                |
|------------------------|--------------------------------|
| Required?              | true                           |
| Default Value          |                                |
| Accept Pipeline Input? | true (ByValue, ByPropertyName) |

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

None

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

### UnknownObject

One of the specified objects was not found.

### ScopeNotFound

One of the specified scopes was not found.

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

### PermissionDenied

You do not have permission to execute this command with the specified objects or scopes.

### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

### CommunicationError

There was a problem communicating with the remote service.

### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Add-HypHypervisorConnectionScope Finance -HypervisorConnectionUid 6702C5D0-C073-4080-A0EE-EC74CB537C52
Adds a single HypervisorConnection to the 'Finance' scope.
```

----- **EXAMPLE 2** -----

```
c:\PS>Add-HypHypervisorConnectionScope Finance,Marketing -HypervisorConnectionUid 6702C5D0-C073-4080-A0EE-EC74CB537C52
Adds a single HypervisorConnection to the multiple scopes.
```

----- **EXAMPLE 3** -----

```
c:\PS>Get-HypHypervisorConnection | Add-HypHypervisorConnectionScope Finance
Adds all visible HypervisorConnection objects to the 'Finance' scope.
```

----- **EXAMPLE 4** -----

```
c:\PS>Add-HypHypervisorConnectionScope Finance -HypervisorConnectionName A*
Adds HypervisorConnection objects with a name starting with an 'A' to the 'Finance' scope.
```



# Add-HypMetadata

Jan 04, 2017

Adds metadata to a hypervisor connection or a hosting unit.

## Syntax

```
Add-HypMetadata [-LiteralPath] <String> [-Property] <String> [-Value] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this command to store additional custom data against a hosting unit or hypervisor connection. This data is not used by the Machine Creation Service, and is provided only for consumers of the services to store any data that may be required for their operations. The metadata is returned along with the hypervisor connection or hosting unit that it is assigned to.

## Related topics

[Remove-HypMetadata](#)

## Parameters

### **-LiteralPath**<String>

Specifies the path within a hosting unit provider to the hosting unit or hypervisor connection item to which to add the metadata. The path specified must be in one of the following formats: <drive>:\HostingUnits\<HostingUnitName> or <drive>:\HostingUnits\{<HostingUnit Uid> or <drive>:\Connections\<Connection Name> or <drive>:\Connections\{<Connection Uid>}

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

### **-Property**<String>

Specifies the property name of the metadata to be added. The property must be unique for the item specified by the path.

The property cannot contain any of the following characters \/:#.\*?=<> | []()''''

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-Value**<String>

Specifies the value for the property.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | LocalHost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Input Type

Systemstring You can pipe a string the contains a path to Add-HypMetadata (Path parameter).

### Return Values

Citrix.Host.Sdk.Metadata

Add-HypMetadata returns an object containing the new definition of the metadata.

Property <string>

Specifies the property of the metadata.

Value <string>

Specifies the value of the metadata.

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### InvalidPath

The path provided is not in the required format.

#### HostingUnitMetadataForeignKeyObjectDoesNotExist

The hosting unit supplied in the path does not exist.

#### HypervisorConnectionMetadataForeignKeyObjectDoesNotExist

The hypervisor connection supplied in the path does not exist.

#### HostingUnitMetadataDuplicateObjectExists

Metadata for the specified hosting unit item already exists with the same property name.

#### HypervisorConnectionMetadataDuplicateObjectExists

Metadata for the specified hypervisor connection item already exists with the same property name.

#### MetadataContainerUndefined

The specified path does not reference a hosting unit or a hypervisor connection.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

## PermissionDenied

The user does not have administrative rights to perform this operation.

## ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Add-HypMetadata -LiteralPath XDHyp:\Connections\MyConnection -Property MyProperty -Value MyValue
```

| Property | Value |
|----------|-------|
|----------|-------|

-----

|            |         |
|------------|---------|
| MyProperty | MyValue |
|------------|---------|

The command adds the metadata with the property name of "MyProperty" and value of "MyValue" to the hypervisor connection item called "MyConnection".

### ----- EXAMPLE 2 -----

```
c:\PS>dir xdhyp\connections\Citrix* | Add-HypMetadata -Property MyProperty -Value MyValue
```

| Property | Value |
|----------|-------|
|----------|-------|

-----

|            |         |
|------------|---------|
| MyProperty | MyValue |
|------------|---------|

|            |         |
|------------|---------|
| MyProperty | MyValue |
|------------|---------|

|            |         |
|------------|---------|
| MyProperty | MyValue |
|------------|---------|

The command adds the metadata with the property name of "MyProperty" and value of "MyValue" to all the hypervisor connection items that begin with the string "Citrix".

# Get-HypConfigurationDataForItem

Jan 04, 2017

Retrieves the configuration data for an item in the Host Service provider path. Note: For this release, only VM items are supported for this operation.

## Syntax

```
Get-HypConfigurationDataForItem [-LiteralPath] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This command provides a mechanism for retrieving extra data about an entry in the hosting unit service provider. The referenced item must be contained within the connections directory in the provider (i.e. XDHyp:\Connections).

This mechanism is used for obtaining data that is not required frequently and/or has a high overhead associated with its retrieval (so as to maintain the responsiveness of the provider). For this release of the PowerShell snap-in, the only provider items that can be used with this operation are VM items. For a VM, this provides a mechanism to obtain the number of CPUs, the amount of Memory (in MB) and hard disk drive capacity (in GB).

## Related topics

Get-Item

## Parameters

**-LiteralPath**<String>

Specifies the path within a hosting unit provider to the item for which configuration data is to be retrieved. The path specified must be in one of the following formats; <drive>:\Connections\<Connection Name>\<Item Path of VM object> or <drive>:\Connections\{<connection Uid>\<Item Path of VM object>} or <drive>:\HostingUnits\<HostingUnit Name>\<Item Path of VM object> or <drive>:\HostingUnits\{<hostingUnit Uid>\<Item Path of VM object>}

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Input Type

System.string You can pipe a string that contains a path to Get-HypConfigurationDataForItem

## Return Values

PSObject

Get-HypConfigurationDataForItem returns a PSObject containing the properties that are appropriate for the item specified by the path.

## Properties for VM Item

-----

CPUCount <int>

Specifies the number of CPUs assigned to the VM.

MemoryMB <int>

The amount of memory allocated to the VM.

HardDiskSizeGB <int>

The capacity of the primary hard drive assigned to the VM.

## Notes

For this release, this cmdlet provides only configuration data for VM objects in the provider. Using a path to an item that is not a VM results in an error.

In the case of failure the following errors can result.

## Error Codes

-----

InputHypervisorItemPathInvalid

The path provided is not to an item in a sub-directory of a connection item or a hosting unit item.

InvalidHypervisorItemPath

No item exists with the specified path.

InvalidHypervisorItem

The item specified by the path exists, but is not a VM Item.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

CommunicationError

An error occurred while communicating with the service.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

HypervisorPermissionDenied

The hypervisor login used does not provide authorization to access the data for this item.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
C:\PS>Get-HypConfigurationDataForItem -LiteralPath XDHyp:\Connections\MyConnection\MyVm.vm
```

| CpuCount | MemoryMB | HardDiskSizeGB |
|----------|----------|----------------|
| 1        | 1024     | 24             |

This command gets the configuration properties for a VM called 'MyVm.vm' within a hypervisor connection called 'MyConnection'.

----- **EXAMPLE 2** -----

```
XDHyp:\HostingUnits\PS>Get-HypConfigurationDataForItem -LiteralPath .\MyVm.vm
```

| CpuCount | MemoryMB | HardDiskSizeGB |
|----------|----------|----------------|
| 1        | 1024     | 24             |

This command gets the configuration properties for a VM called 'MyVm.vm' within the current directory. The dot (.) represents the current location (not its contents).

----- **EXAMPLE 3** -----

```
C:\PS>(Get-HypConfigurationDataForItem -LiteralPath XDHyp:\Connections\MyConnection\MyVm.vm).CPUCount
```

This command gets the CPU count for a VM called 'MyVm.vm'. The CPUCount is just one property of the VM items. To see all properties of an item, type "(Get-HypConfigurationDataForItem <ItemPath>) | Get-Member".



# Get-HypConfigurationObjectForItem

Jan 04, 2017

Retrieves the configuration data for an item in the Host Service provider path. Note: For this release, only VM items are supported for this operation.

## Syntax

```
Get-HypConfigurationObjectForItem [-LiteralPath] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This command provides a mechanism for retrieving extra data about an entry in the hosting unit service provider. The referenced item must be contained within the connections directory in the provider (i.e. XDHyp:\Connections).

This mechanism is used for obtaining data that is not required frequently and/or has a high overhead associated with its retrieval (so as to maintain the responsiveness of the provider). For this release of the PowerShell snap-in the only provider items that can be used with this operation are VM items. For a VM this provides a mechanism to obtain the number of CPUs, the amount of Memory (in MB) and hard disk drive capacity (GB).

## Related topics

Get-Item

## Parameters

**-LiteralPath**<String>

Specifies the path within a hosting unit provider to the item for which configuration data is to be retrieved. The path specified must be in one of the following formats; <drive>:\Connections\<Connection Name>\<Item Path of VM object> or <drive>:\Connections\{<connection Uid>\<Item Path of VM object>} or <drive>:\HostingUnits\<HostingUnit Name>\<Item Path of VM object> or <drive>:\HostingUnits\{<hostingUnit Uid>\<Item Path of VM object>}

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in will connect to. This can be provided as a host name or an IP address.

|               |                                                                                        |
|---------------|----------------------------------------------------------------------------------------|
| Required?     | false                                                                                  |
| Default Value | LocalHost. Once a value is provided by any cmdlet, this value will become the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Input Type

System.String You can pipe a string that contains a path to Get-HypConfigurationDataForItem

## Return Values

### PSObject

Get-HypConfigurationDataForItem returns a PSObject containing the properties that are appropriate for the item specified by the path.

### Properties for VM Item

-----

#### CPUCount <int>

Specifies the number of CPUs assigned to the VM.

#### MemoryMB <int>

The amount of memory allocated to the VM.

#### HardDiskSizeGB <int>

The capacity of the primary hard drive assigned to the VM.

#### Network Map

The networks that this VM or Snapshot is connected to

## Notes

For this release this cmdlet only provides configuration data for VM objects in the provider. Using a path to an item that is not a VM will result in an error.

In the case of failure the following errors can be produced.

### Error Codes

-----

#### InputHypervisorItemPathInvalid

The path provided is not to an item in a sub directory of a connection item or a hosting unit item.

#### InvalidHypervisorItemPath

No item exists with the specified path.

#### InvalidHypervisorItem

The item specified by the path exists, but is not a VM Item.

#### DatabaseError

An error occurred in the service whilst attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed as the database for the service is not configured.

#### DataStoreException

An error occurred in the service whilst attempting a database operation - communication to database failed for various reasons.

#### CommunicationError

An error occurred whilst communicating with the service.

#### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

#### HypervisorPermissionDenied

The hypervisor login used does not provide authorization to access the data for this item.

#### ExceptionThrown

An unexpected error occurred. To locate more details see the Windows event logs on the controller being used, or examine the XenDesktop logs.

### Examples

#### ----- EXAMPLE 1 -----

```
C:\PS>Get-HypConfigurationDataForItem -LiteralPath XDHyp:\Connections\MyConnection\MyVm.vm
```

| CpuCount | MemoryMB | HardDiskSizeGB |
|----------|----------|----------------|
| 1        | 1024     | 24             |

This command gets the configuration properties for a VM called 'MyVm.vm' within a hypervisor connection called 'MyConnection'.

#### ----- EXAMPLE 2 -----

```
XDHyp\HostingUnits\PS>Get-HypConfigurationDataForItem -LiteralPath .\MyVm.vm
```

| CpuCount | MemoryMB | HardDiskSizeGB |
|----------|----------|----------------|
| 1        | 1024     | 24             |

This command gets the configuration properties for a VM called 'MyVm.vm' within the current directory. The dot (.) represents the current location (not its contents).

----- **EXAMPLE 3** -----

```
C:\PS>(Get-HypConfigurationDataForItem -LiteralPath XDHyp:\Connections\MyConnection\MyVm.vm).CPUCount
```

This command gets the CPU count for a VM called 'MyVm.vm'. The CPUCount is just one property of the VM items. To see all properties of an item, type "(Get-HypConfigurationDataForItem <ItemPath> | Get-Member)".

# Get-HypConnectionRegion

Jan 04, 2017

Enumerates the regions of a hypervisor connection that are based on cloud technology.

## Syntax

```
Get-HypConnectionRegion [-LiteralPath] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this command to enumerate the available regions within a public or private cloud, when making hypervisor connections to cloud services. Sometimes, regions need to be selected and applied before the cloud connection can be used in a meaningful way. This cmdlet allows the supported regions to be listed so that one may be selected.

Once a region has been chosen, use the standard Set-Item provider cmdlet to select it. See the examples for further details.

This cmdlet may return no output, in which case the cloud connection can be considered "regionless" (or, implicitly, all within a single region). In such cases, there is no need to select a region, and the hypervisor connection can be used as is.

## Related topics

Set-Item

## Parameters

### **-LiteralPath**<String>

Specifies the path to the hypervisor connection whose regions are being examined. This cmdlet is valid only for hypervisor connections that have the UsesCloudInfrastructure flag set to true. The path must be in one of the following formats: <drive>:\Connections\<ConnectionName> or <drive>:\Connections\{<Connection Uid>}

|                        |                |
|------------------------|----------------|
| Required?              | true           |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | localhost. When a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Input Type

Systemstring You can pipe a string that contains a path to Get-HypConnectionRegion (LiteralPath parameter).

## Return Values

Citrix.Host.Sdk.HypervisorRegion

Get-HypConnectionRegion returns zero or more instances of the HypervisorConnectionRegion object, each of which contain the following properties:

Name <string> Specifies the unique name of the region. Endpoint <string> Specifies the URL endpoint that is specific to the region, if relevant. This may be an empty string, and is returned only for information purposes.

A full list of the hypervisor networks that are exposed for use in the hosting unit.

## Notes

In the case of failure, the following errors can be produced.

### Error Codes

-----

#### ConnectionsPathInvalid

The path provided is not to an item in the Connections subdirectory of the host service provider.

#### HypervisorConnectionObjectNotFound

The path provided could not be resolved to an existing hypervisor connection. The name or GUID is invalid.

#### HypervisorInMaintenanceMode

The hypervisor for the connection is in maintenance mode.

#### ConnectionIsNotCloud

The hypervisor connection is not associated with cloud infrastructure, making it invalid to enumerate regions.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation. Communication with the database failed for various reasons.

## CommunicationError

An error occurred while communicating with the service.

## PermissionDenied

The user does not have administrative rights to perform this operation.

## ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used, or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-HypConnectionRegions -LiteralPath XDHyp:\Connections\AWS
```

```
RegionName      : us-east-1  
Endpoint        : ec2.us-east-1.amazonaws.com
```

```
RegionName      : us-west-1  
Endpoint        : ec2.us-west-1.amazonaws.com
```

```
RegionName      : eu-west-1  
Endpoint        : ec2.eu-west-1.amazonaws.com
```

```
(...)
```

```
c:\PS>Set-Item -Path XDHyp:\Connections\AWS -Region "us-east-1"
```

This sequence of commands enumerates the available regions of an Amazon AWS cloud connection, and then selects one of them for use in the connection.

# Get-HypDBConnection

Jan 04, 2017

Gets the database string for the specified data store used by the Host Service.

## Syntax

```
Get-HypDBConnection [-AdminAddress <String>][<CommonParameters>]
```

## Detailed Description

Returns the database connection string for the specified data store.

If the returned string is blank, no valid connection string has been specified. In this case the service is running, but is idle and awaiting specification of a valid connection string.

## Related topics

[Get-HypServiceStatus](#)

[Set-HypDBConnection](#)

[Test-HypDBConnection](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

system.string

The database connection string configured for the Host Service.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----



NoDBConnections

The database connection string for the Host Service has not been specified.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-HypDBConnection
```

Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True  
Get the database connection string for the Host Service.

# Get-HypDBSchema

Jan 04, 2017

Gets a script that creates the Host Service database schema for the specified data store.

## Syntax

```
Get-HypDBSchema [-DatabaseName <String>] [-ServiceGroupName <String>] [-ScriptType <ScriptTypes>] [-LocalDatabase] [-Sid <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets SQL scripts that can be used to create a new Host Service database schema, add a new Host Service to an existing site, remove a Host Service from a site, or create a database server logon for a Host Service. If no Sid parameter is provided, the scripts obtained relate to the currently selected Host Service instance, otherwise the scripts relate to Host Service instance running on the machine identified by the Sid provided. When obtaining the Evict script, a Sid parameter must be supplied. The current service instance is that on the local machine, or that explicitly specified by the last usage of the -AdminAddress parameter to a Host SDK cmdlet. The service instance used to obtain the scripts does not need to be a member of a site or to have had its database connection configured. The database scripts support only Microsoft SQL Server, or SQL Server Express, and require Windows integrated authentication to be used. They can be run using SQL Server's SQLCMD utility, or by copying the script into an SQL Server Management Studio (SSMS) query window and executing the query. If using SSMS, the query must be executed in 'SMDCMD mode'. The ScriptType parameter determines which script is obtained. If ScriptType is not specified, or is FullDatabase, the script contains:

- o Creation of service schema
- o Creation of database server logon
- o Creation of database user
- o Addition of database user to Host Service roles

If ScriptType is Instance, the returned script contains:

- o Creation of database server logon
- o Creation of database user
- o Addition of database user to Host Service roles

If ScriptType is Evict, the returned script contains:

- o Removal of Host Service instance from database
- o Removal of database user

If ScriptType is Login, the returned script contains:

- o Creation of database server logon only

If the service uses two data stores they can exist in the same database. You do not need to configure a database before using this command.

## Related topics

[Set-HypDBConnection](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database for which the schema will be generated.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ServiceGroupName**<String>

Specifies the name of the service group to be used when creating the database schema. The service group is a collection of all the Host services that share the same database instance and are considered equivalent; that is, all the services within a service group can be used interchangeably.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Script Type**<ScriptTypes>

Specifies the type of database script returned. Available script types are:

### Database

Returns a full database script that can be used to create a database schema for the Host Service in a database instance that does not already contain a schema for this service. The DatabaseName and ServiceGroupName parameters must be specified to create a script of this type.

### Instance

Returns a permissions script that can be used to add further Host services to an existing database instance that already contains the full Host service schema, associating the services to the Service Group. The Sid parameter can optionally be specified to create a script of this type.

### Login

Returns a database logon script that can be used to add the required logon accounts to an existing database instance that contains the Host Service schema. This is used primarily when creating a mirrored database environment. The

DatabaseName parameter must be specified to create a script of this type.

#### Evict

Returns a script that can be used to remove the specified Host Service from the database entirely. The DatabaseName and Sid parameters must be specified to create a script of this type.

|                        |          |
|------------------------|----------|
| Required?              | false    |
| Default Value          | Database |
| Accept Pipeline Input? | false    |

#### -LocalDatabase<SwitchParameter>

Specifies whether the database script is to be used in a database instance run on the same controller as other services in the service group. Including this parameter ensures the script creates only the required permissions for local services to access the database schema for Host services.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

#### -Sid<String>

Specifies the SID of the controller on which the Host Service instance to remove from the database is running.

|                        |                |
|------------------------|----------------|
| Required?              | false          |
| Default Value          |                |
| Accept Pipeline Input? | true (ByValue) |

#### -AdminAddress<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

### Systemstring

A string containing the required SQL script for application to a database.

## Notes

The scripts returned support Microsoft SQL Server Express Edition, Microsoft SQL Server Standard Edition, and Microsoft SQL Server Enterprise Edition databases only, and are generated on the assumption that integrated authentication will be used.

If the ScriptType parameter is not included or set to 'FullDatabase', the full database script is returned, which will:

Create the database schema.

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist).

If the ScriptType parameter is set to 'Instance', the script will:

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist) and associate it with a user.

If the ScriptType parameter is set to 'Login', the script will:

Create the logon (providing the schema does not already exist) and associate it with a pre-existing user of the same name.

If the LocalDatabase parameter is included, the NetworkService account will be added to the list of accounts permitted to access the database. This is required only if the database is run on a controller.

If the command fails, the following errors can be returned.

## Error Codes

-----

### GetSchemasFailed

The database schema could not be found.

### ActiveDirectoryAccountResolutionFailed

The specified Active Directory account or Group could not be found.

### DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-HypDBSchema -DatabaseName MyDB -ServiceGroupName MyServiceGroup > c:\HypSchema.sql  
Get the full database schema for site data store of the Host Service and copy it to a file called 'c:\HypSchema.sql'.
```

This script can then be used to create the schema in a pre-existing database named 'MyDB' that does not already contain a Host Service site schema.

### ----- EXAMPLE 2 -----

```
c:\PS>Get-HypDBSchema -DatabaseName MyDB -scriptType Login > c:\HostLogins.sql  
Get the logon scripts for the Host Service.
```

# Get-HypDBVersionChangeScript

Jan 04, 2017

Gets a script that updates the Host Service database schema.

## Syntax

```
Get-HypDBVersionChangeScript -DatabaseName <String> -TargetVersion <Version> [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Returns a database script that can be used to upgrade or downgrade the site or secondary schema for the Host Service from the current schema version to a different version.

## Related topics

[Get-HypInstalledDBVersion](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database instance to which the update applies.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-TargetVersion**<Version>

Specifies the version of the database you want to update to.

|                        |       |
|------------------------|-------|
| Required?              | true  |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

System.Management.Automation.PSObject

A PSObject containing the required SQL script for application to a database.

## Notes

The PSObject returned by this cmdlet contains the following properties:

-- Script The raw text of the SQL script to apply the update, or null in the case when no upgrade path to the specified target version exists.

-- NeedExclusiveAccess Indicates whether all services in the service group must be shut down during the update or not.

-- CanUndo Indicates whether the generated script allows the updated schema to be reverted to the state prior to the update.

Scripts to update the schema version are stored in the database so any service in the service group can obtain these scripts. Extreme caution should be exercised when using update scripts. Citrix recommends backing up the database before attempting to upgrade the schema. Database update scripts may require exclusive use of the schema and so may not be able to execute while any Host services are running. However, this depends on the specific update being carried out.

After a schema update has been carried out, services that require the previous version of the schema may cease to operate. The ServiceState parameter reported by the Get-HypServiceStatus command provides information about service compatibility. For example, if the schema has been upgraded to a more recent version that a service cannot use, the service reports "DBNewerVersionThanService".

If the command fails, the following errors can be returned.

## Error Codes

-----

### NoOp

The operation was successful but had no effect.

### NoDBConnections

The database connection string for the Host Service has not been specified.

### DatabaseError

An error occurred in the service while attempting a database operation.



## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $update = Get-HypDBVersionChangeScript -DatabaseName MyDb -TargetVersion 1.0.75.0
```

```
C:\PS> $update.Script > update_75.sql
```

Gets an SQL update script to update the current schema to version 1.0.75.0. The resulting update\_75.sql script is suitable for direct use with the SQL Server SQLCMD utility.

# Get-HypHypervisorPlugin

Jan 04, 2017

Gets the available hypervisor types.

## Syntax

```
Get-HypHypervisorPlugin [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this command to retrieve a list of all the available hypervisor types, and their localized names.

## Related topics

New-Item

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | LocalHost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

## Return Values

Citrix.Host.Sdk.HypervisorPlugin

Get-HypHypervisorPlugin returns a list of objects containing the definition of the hypervisor plug-ins.

ConnectionType <Citrix.XDIInterServiceTypes.ConnectionType>

The hypervisor connection type. This can be one of the following:

XenServer - XenServer hypervisor

SCVMM - Microsoft SCVMM/Hyper-V

vCenter - VMWare vSphere/ESX

Custom - a third-party hypervisor

DisplayName <string>

The localized display name (localized using the locale of the Powershell snap-in session)

PluginFactoryName <string>

The name of the hypervisor plug-in factory used to manage the hypervisor connections.

## Notes

To use third-party plug-ins, the plug-in assemblies must be installed into the appropriate location on each controller machine that forms part of the Citrix controller site. Failure to do this can result in unpredictable behavior, especially during service failover conditions.

In the case of failure the following errors can result.

### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### CommunicationError

An error occurred while communicating with the service.

#### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS> Get-HypHypervisorPlugin | Format-Table -AutoSize
```

| ConnectionType | DisplayName              | PluginFactoryName  |
|----------------|--------------------------|--------------------|
| SCVMM          | Microsoft virtualization | MicrosoftPSFactory |
| VCenter        | VMware virtualization    | VmwareFactory      |
| XenServer      | Citrix XenServer         | XenFactory         |

Get the available hypervisor management plug-ins.

# Get-HypInstalledDBVersion

Jan 04, 2017

Gets a list of all available database schema versions for the Host Service.

## Syntax

```
Get-HypInstalledDBVersion [-Upgrade] [-Downgrade] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the current version of the Host Service database schema, if no flags are set, otherwise returns versions for which upgrade or downgrade scripts are available and have been stored in the database.

## Related topics

## Parameters

**-Upgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be updated should be returned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Downgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be reverted should be returned.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| Required?     | false                                                                              |
| Default Value | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |

|                        |       |
|------------------------|-------|
| Accept Pipeline Input? | false |
|------------------------|-------|

## Return Values

### System.Version

The Get-HypInstalledDbVersion command returns objects containing the new definition of the Host Service database schema version.

Major <Integer>

Minor <Integer>

Build <Integer>

Revision <Integer>

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

Both the Upgrade and Downgrade flags were specified.

#### NoOp

The operation was successful but had no effect.

#### NoDBConnections

The database connection string for the Host Service has not been specified.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-HypInstalledDBVersion
```

```
Major Minor Build Revision
```

```
-----
```

```
5 6 0 0
```

Get the currently installed version of the Host Service database schema.

### ----- EXAMPLE 2 -----

```
c:\PS>Get-HypInstalledDBVersion -Upgrade
```

```
Major Minor Build Revision
```

```
-----
```

```
6 0 0 0
```

Get the versions of the Host Service database schema for which upgrade scripts are supplied.

# Get-HypScopedObject

Jan 04, 2017

Gets the details of the scoped objects for the Host Service.

## Syntax

```
Get-HypScopedObject [-ScopeId <Guid>] [-ScopeName <String>] [-ObjectType <ScopedObjectType>] [-ObjectId <String>] [-ObjectName <String>] [-Description <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns a list of directly scoped objects including the names and identifiers of both the scope and object as well as the object description for display purposes.

There will be at least one result for every directly scoped object. When an object is associated with multiple scopes the output contains one result per scope duplicating the object details.

No records are returned for the All scope, though if an object is not in any scope a result with a null ScopeId and ScopeName will be returned.

## Related topics

### Parameters

#### **-ScopeId**<Guid>

Gets scoped object entries for the given scope identifier.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

#### **-ScopeName**<String>

Gets scoped object entries with the given scope name.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ObjectType**<ScopedObjectType>

Gets scoped object entries for objects of the given type.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ObjectId**<String>

Gets scoped object entries for objects with the specified object identifier.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-ObjectName**<String>

Gets scoped object entries for objects with the specified object identifier.

|                        |                       |
|------------------------|-----------------------|
| Required?              | false                 |
| Default Value          |                       |
| Accept Pipeline Input? | true (ByPropertyName) |

**-Description**<String>

Gets scoped object entries for objects with the specified description.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-Property**<String[]>



Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Hyp\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | False |
| Accept Pipeline Input? | false |

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 250   |
| Accept Pipeline Input? | false |

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          | 0     |
| Accept Pipeline Input? | false |

### **-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| Required?              | false                                                   |
| Default Value          | The default sort order is by name or unique identifier. |
| Accept Pipeline Input? | false                                                   |

### **-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Hyp\_Filtering for details.

|                        |       |
|------------------------|-------|
| Required?              | false |
| Default Value          |       |
| Accept Pipeline Input? | false |

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

|                        |                                                                                    |
|------------------------|------------------------------------------------------------------------------------|
| Required?              | false                                                                              |
| Default Value          | Localhost. Once a value is provided by any cmdlet, this value becomes the default. |
| Accept Pipeline Input? | false                                                                              |

### Return Values

Citrix.Host.Sdk.ScopedObject

The Get-HypScopedObject command returns an object containing the following properties:

ScopeId <Guid?>

Specifies the unique identifier of the scope.

ScopeName <String>

Specifies the display name of the scope.

ObjectType <ScopedObjectType>

Type of the object this entry relates to.

ObjectId <String>

Unique identifier of the object.

ObjectName <String>

Display name of the object

Description <String>

Description of the object (possibly \$null if the object type does not have a description).

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

UnknownObject

One of the specified objects was not found.

PartialData

Only a subset of the available data was returned.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

CouldNotQueryDatabase

The query required to get the database was not defined.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-HypScopedObject -ObjectType Scheme
```

```
ScopelId   : eff6f464-f1ee-4442-add3-99982e0cec01
ScopeName  : Sales
ObjectType : Scheme
ObjectId   : cd4174ee-9e4b-4e57-b126-9dbf757fe493
ObjectName : MyExampleScheme
Description : Test scheme
```

```
ScopelId   : 304e0fa7-d390-47f0-a94f-7e956a324c41
ScopeName  : Finance
ObjectType : Scheme
ObjectId   : cd4174ee-9e4b-4e57-b126-9dbf757fe493
ObjectName : MyExampleScheme
Description : Test scheme
```

```
ScopelId   :
ScopeName  :
ObjectType : Scheme
ObjectId   : 5062e46b-71bc-4ac9-901a-30fe6797e2f6
ObjectName : AnotherScheme
Description : Another scheme in no scopes
```

Gets all of the scoped objects with type Scheme. The example output shows a scheme object (MyExampleScheme) in two scopes Sales and Finance, and another scheme (AnotherScheme) that is not in any scope. The ScopelId and ScopeName values returned are null in the final record.

# Get-HypService

Jan 04, 2017

Gets the service record entries for the Host Service.

## Syntax

```
Get-HypService [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns instances of the Host Service that the service publishes. The service records contain account security identifier information that can be used to remove each service from the database.

A database connection for the service is required to use this command.

## Related topics

### Parameters

#### **-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: `-Metadata "abc:x"` matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

#### **-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

#### **-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Hyp\_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-Sort By**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

### **-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Hyp\_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

Citrix.Host.Sdk.Service

The Get-HypServiceInstance command returns an object containing the following properties.

Uid <Integer>

Specifies the unique identifier for the service in the group. The unique identifier is an index number.

ServiceHostId <Guid>

Specifies the unique identifier for the service instance.

DNSName <String>

Specifies the domain name of the host on which the service runs.

MachineName <String>

Specifies the short name of the host on which the service runs.

CurrentState <Citrix.Fma.Sdk.ServiceCore.ServiceState>

Specifies whether the service is running, started but inactive, stopped, or failed.

LastStartTime <DateTime>

Specifies the date and time at which the service was last restarted.

LastActivityTime <DateTime>

Specifies the date and time at which the service was last stopped or restarted.

OSType

Specifies the operating system installed on the host on which the service runs.

OSVersion

Specifies the version of the operating system installed on the host on which the service runs.

ServiceVersion

Specifies the version number of the service instance. The version number is a string that reflects the full build version of the service.

DatabaseUserName <string>

Specifies for the service instance the Active Directory account name with permissions to access the database. This will be either the machine account or, if the database is running on a controller, the NetworkService account.

Sid <string>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

ActiveSiteServices <string[]>

Specifies the names of active site services currently running in the service. Site services are components that perform long-running background processing in some services. This field is empty for services that do not contain site services.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### PartialData

Only a subset of the available data was returned.

#### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

#### CouldNotQueryDatabase

The query required to get the database was not defined.



## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-HypService
```

```
Uid          : 1
ServiceHostId : aef6f464-f1ee-4042-a523-66982e0cecd0
DNSName      : MyServer.company.com
MachineName  : MYSERVER
CurrentState  : On
LastStartTime : 04/04/2011 15:25:38
LastActivityTime : 04/04/2011 15:33:39
OSType       : Win32NT
OSVersion    : 6.1.7600.0
ServiceVersion : 5.1.0.0
DatabaseUserName : NT AUTHORITY\NETWORK SERVICE
SID          : S-1-5-21-2316621082-1546847349-2782505528-1165
ActiveSiteServices : {MySiteService1, MySiteService2...}
Get all the instances of the Host Service running in the current service group.
```

# Get-HypServiceAddedCapability

Jan 04, 2017

Gets any added capabilities for the Host Service on the controller.

## Syntax

```
Get-HypServiceAddedCapability [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables updates to the Host Service on the controller to be detected.

You do not need to configure a database connection before using this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

System.String

List containing added capabilities.

String containing added capabilities.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-HypServiceAddedCapability  
Get the added capabilities of the Host Service.
```

# Get-HypServiceInstance

Jan 04, 2017

Gets the service instance entries for the Host Service.

## Syntax

```
Get-HypServiceInstance [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns service interfaces published by the instance of the Host Service. Each instance of a service publishes multiple interfaces with distinct interface types, and each of these interfaces is represented as a ServiceInstance object. Service instances can be used to register the service with a central configuration service so that other services can use the functionality.

You do not need to configure a database connection to use this command.

## Related topics

### Parameters

**-AdminAddress** <String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.Host.Sdk.ServiceInstance

The Get-HypServiceInstance command returns an object containing the following properties.

ServiceGroupUid <Guid>

\n Specifies the unique identifier for the service group of which the service is a member.

ServiceGroupName <String>

\n Specifies the name of the service group of which the service is a member.

ServiceInstanceUID <Guid>

\n Specifies the unique identifier for registered service instances, which are service instances held by and obtained from a

central configuration service. Unregistered service instances do not have unique identifiers.

ServiceType <String>

\n Specifies the service instance type. For this service, the service instance type is always Hyp.

Address

\n Specifies the address of the service instance. The address can be used to access the service and, when registered in the central configuration service, can be used by other services to access the service.

Binding

\n Specifies the binding type that must be used to communicate with the service instance. In this release of XenDesktop, the binding type is always 'wcf\_HTTP\_kerb'. This indicates that the service provides a Windows Communication Foundation endpoint that uses HTTP binding with integrated authentication.

Version

\n Specifies the version of the service instance. The version number is used to ensure that the correct versions of the services are used for communications.

ServiceAccount <String>

\n Specifies the Active Directory account name for the machine on which the service instance is running. The account name is used to provide information about the permissions required for interservice communications.

ServiceAccountSid <String>

\n Specifies the Active Directory account security identifier for the machine on which the service instance is running.

InterfaceType <String>

\n Specifies the interface type. Each service can provide multiple service instances, each for a different purpose, and the interface defines the purpose. Available interfaces are:

\n SDK - for PowerShell operations

\n InterService - for operations between different services

\n Peer - for communications between services of the same type

Metadata <Citrix.Host.Sdk.Metadata[]>

The collection of metadata associated with registered service instances, which are service instances held by and obtained from a central configuration service. Metadata is not stored for unregistered service instances.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-HypServiceInstance
```

```
Address      : http://MyServer.com:80/Citrix/SdkHostingUnitService
Binding      : wcf_HTTP_kerb
InterfaceType : SDK
Metadata     :
MetadataMap  :
ServiceAccount : ENG\MyAccount$
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164
ServiceGroupName : MyServiceGroup
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000
ServiceType  : Hyp
Version      : 1
```

```
Address      : http://MyServer.com:80/Citrix/SdkHostingUnitService/IServiceApi
Binding      : wcf_HTTP_kerb
InterfaceType : InterService
Metadata     :
MetadataMap  :
```

ServiceAccount : ENG\MyAccount  
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164  
ServiceGroupName : MyServiceGroup  
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d  
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000  
ServiceType : Hyp  
Version : 1

Get all instances of the Host Service running on the specified machine. For remote services, use the AdminAddress parameter to define the service for which the interfaces are required. If the AdminAddress parameter has not been specified for the runspace, service instances running on the local machine are returned.

# Get-HypServiceStatus

Jan 04, 2017

Gets the current status of the Host Service on the controller.

## Syntax

```
Get-HypServiceStatus [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables the status of the Host Service on the controller to be monitored. If the service has multiple data stores it will return the overall state as an aggregate of all the data store states. For example, if the site data store status is OK and the secondary data store status is DBUnconfigured then it will return DBUnconfigured.

## Related topics

[Set-HypDBConnection](#)

[Test-HypDBConnection](#)

[Get-HypDBConnection](#)

[Get-HypDBSchema](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Get-HypServiceStatus command returns an object containing the status of the Host Service together with extra diagnostics information.

DBUnconfigured

The Host Service does not have a database connection configured.

DBRejectedConnection



The database rejected the logon attempt from the Host Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

#### InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Host Service schema has not been added to the database.

#### DBNotFound

The specified database could not be located with the configured connection string.

#### DBNewerVersionThanService

The version of the Host Service currently in use is incompatible with the version of the Host Service schema on the database. Upgrade the Host Service to a more recent version.

#### DBOlderVersionThanService

The version of the Host Service schema on the database is incompatible with the version of the Host Service currently in use. Upgrade the database schema to a more recent version.

#### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

#### OK

The Host Service is running and is connected to a database containing a valid schema.

#### Failed

The Host Service has failed.

#### Unknown

(0) The service status cannot be determined.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-HypServiceStatus
```

DBUnconfigured

Get the current status of the Host Service.

# Get-HypVMMacAddress

Jan 04, 2017

Retrieves a list the MAC addresses for the VMs in the specified connection.

## Syntax

```
Get-HypVMMacAddress [-LiteralPath] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this command to obtain a list of MAC addresses of all the virtual machines in the specified connection.

## Related topics

Get-Item

## Parameters

### **-LiteralPath**<String>

The path to a connection item in the hosting provider. Paths to anything other than a connection item will result in an error being returned. The path can be provided as either <drive>:\connections\<Connection Name> or <drive>:\connections\{<Connection Uid>}

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Input Type

Systemstring You can pipe a string that contains a path to Get-HypConfigurationDataForItem

## Return Values

Citrix.Host.Sdk.HypervisorVMObject

Get-HypVMMMacAddress returns an object containing the following properties.

MacAddress <string> - specifies the MAC address of the VM.

VMId <string> - specifies the identifier for the VM as defined in the hypervisor hosting it.

## Notes

The path must refer to a connection item. Hosting unit items are not valid.

In the case of failure, the following errors can result.

## Error Codes

-----

### InputConnectionsPathInvalid

If the path is not provided in an expected format, an InputConnectionsPathInvalid error results.

### HypervisorConnectionObjectNotFound

The hypervisor connection object specified cannot be found.

### HypervisorInMaintenanceMode

The hypervisor for the connection is in maintenance mode.

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

### CommunicationError

An error occurred while communicating with the service.

### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS>Get-HypVMMacAddress -LiteralPath XDHyp:\Connections\MyConnection
```

MacAddress	VMId
-----	----
52:b0:1c:ed:60:fa	f3395d2a-a196-41c2-e37d-764acf871599
62:6f:f0:40:d5:af	5f4457b0-cc3c-f806-8ca7-5f57e4bdf2d1
4e:a5:9f:00:b2:0c	3115177b-85a9-d8ee-d0f9-0c7437483c09

This command gets the MAC addresses for the connection called "MyConnection".

### ----- EXAMPLE 2 -----

```
XDHyp:\Connections\MyConnection>Get-HypVMMacAddress -Path .
```

MacAddress	VMId
-----	----
52:b0:1c:ed:60:fa	f3395d2a-a196-41c2-e37d-764acf871599
62:6f:f0:40:d5:af	5f4457b0-cc3c-f806-8ca7-5f57e4bdf2d1
4e:a5:9f:00:b2:0c	3115177b-85a9-d8ee-d0f9-0c7437483c09

This command gets the MAC addresses for the connection at the current directory. The dot (.) represents the current location (not its contents).

### ----- EXAMPLE 3 -----

```
C:\PS>Get-HypVMMacAddress -LiteralPath Xdhyp:\connections\"{268c66db-9b8c-47f6-9265-42326dbff006}"
```

This command gets the MAC addresses for the connection that has a ConnectionUid of 268c66db-9b8c-47f6-9265-42326dbff006.

# Get-HypVolumeServiceConfiguration

Jan 04, 2017

Gets instances of the VolumeServiceConfiguration that are configured for this site.

## Syntax

```
Get-HypVolumeServiceConfiguration [[-VolumeServiceConfigurationName] <String>] [-VolumeServiceConfigurationUid <Guid>] [-ConnectionType <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Retrieve VolumeServiceConfigurations whose properties match the given filter criteria. If no parameters are specified, this cmdlet retrieves all VolumeServiceConfiguration objects.

## Related topics

[Set-HypVolumeServiceConfiguration](#)

## Parameters

**-VolumeServiceConfigurationName**<String>

Specifies a filter for the configuration name.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-VolumeServiceConfigurationUid**<Guid>

Specifies a filter for the configuration ID.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ConnectionType**<String>

Specifies a filter for the cloud connection type, such as "AWS" or "CloudPlatform".

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Hyp\_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

**-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Hyp\_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Input Type



string The name of a configuration (or set of configurations) to retrieve.

## Return Values

Citrix.Host.Sdk.VolumeServiceConfiguration

This cmdlet returns matching VolumeServiceConfiguration objects.

## Examples

### ----- **EXAMPLE 1** -----

```
C:\PS>Get-HypVolumeServiceConfiguration -VolumeServiceConfigurationName SiteDefault -ConnectionType CloudPlatform
```

This command lists the site default volume service configuration for connections that use Citrix Cloud Platform.

# Get-HypXenServerAddress

Jan 04, 2017

Gets all the available addresses for a XenServer hypervisor connection.

## Syntax

```
Get-HypXenServerAddress [-LiteralPath] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this cmdlet to retrieve all the available hypervisor connection addresses that can be used to connect to the same XenServer pool. When used in conjunction with `Add-HypHypervisorAddress`, you can easily populate a connection with all the addresses that can be used to provide full failover support for a XenServer connection.

If the addresses are https addresses, the command uses the certificates installed on the XenServers to provide suitable https addresses where possible. Only servers that can be resolved are returned.

## Related topics

[Add-HypHypervisorConnectionAddress](#)

## Parameters

### **-LiteralPath**<String>

Specifies the path within a Host Service provider to the hypervisor connection item to which to add the address. The path specified must be in one of the following formats: <drive>:\Connections\<HypervisorConnectionName> or <drive>:\Connections\{HHypervisorConnection Uid}

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Systemstring

`Get-HypXenServerAddress` returns a list of strings containing the available address.

## Notes

For this to work as required with https connections, the certificates installed on the XenServers must be trusted by all controllers. Typically this means having the root certificate for the certificate trust chain installed on all controllers. Wildcard certificates are not supported for this operation.

In the case of failure, the following errors can result.

## Error Codes

-----

### InputConnectionsPathInvalid

The path provided is not to an item in a sub-directory of a hosting unit item.

### HypervisorConnectionNotXenServer

The hypervisor connection to which the path refers is not for a Citrix XenServer hypervisor.

#### HypervisorConnectionObjectNotFound

The hypervisor connection at the path specified cannot be located.

#### HypervisorInMaintenanceMode

The hypervisor for the connection is in maintenance mode.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
c:\PS>Get-HypXenServerAddress -Path XDHyp:\Connections\MyConnection
```

```
https://myserver.com
```

```
https://myServer1.com
```

Gets the available addresses for the connection "MyConnection".

##### ----- EXAMPLE 2 -----

```
c:\PS>Get-HypXenServerAddress -LiteralPath XDHyp:\Connections\MyConnection | Add-HypHypervisorConnectionAddress -Path XDHyp:\Connections\MyConnectionPath
```

Adds all the available addresses for the XenServer pool in "MyConnection" to the hypervisor connection.

# Grant-HypSecurityGroupEgress

Jan 04, 2017

Adds an egress rule to a security group.

## Syntax

```
Grant-HypSecurityGroupEgress [-LiteralPath <String> -GroupId <String[]> -Protocol <String> [-FromPort <Decimal>] [-ToPort <Decimal>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Grant-HypSecurityGroupEgress [-LiteralPath <String> -IPRange <String[]> -Protocol <String> [-FromPort <Decimal>] [-ToPort <Decimal>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Adding an egress rule permits network traffic from instances within the security group to pass to one or more destination CIDR IP address ranges or security groups.

## Related topics

Amazon AuthorizeSecurityGroupEgress: <http://docs.aws.amazon.com/AWSEC2/latest/APIReference/ApiReference-query-AuthorizeSecurityGroupEgress.html>

IANA protocol numbers: <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>

[Grant-HypSecurityGroupIngress](#)

[Revoke-HypSecurityGroupIngress](#)

[Revoke-HypSecurityGroupEgress](#)

## Parameters

### **-LiteralPath**<String>

Specifies the full XD Hyp provider path to the security group, equivalent to the FullPath property of the security group object. The path can specify a security group relative to a hypervisor connection or hosting unit.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

### **-Protocol**<String>

Specifies the protocol name or number. Protocol numbers can be found at: <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>

Use -1 to specify all protocols.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-GroupId**<String[]>

Specifies one or more destination security groups to which traffic will be permitted by this rule. This parameter cannot be specified in conjunction with IPRange.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-IPRange**<String[]>

Specifies one or more destination CIDR IP address ranges to which traffic will be permitted by this rule. This parameter cannot be specified in conjunction with IPRange.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-FromPort**<Decimal>

The start of the port range for port based protocols. For ICMP this specifies the type number.

Use -1 to specify all ICMP types.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-ToPort**<Decimal>

The end of the port range for port based protocols. For ICMP this specifies the type number, where -1 can be used to specify all ICMP types.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and

Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Input Type**

System.string The LiteralPath can be piped in.

**Return Values**

None

**Notes**

Security groups can be added and removed using the New-Item and Remove-Item cmdlets.

**Examples**

----- **EXAMPLE 1** -----

```
c:\PS> $Group = New-Item -ItemType SecurityGroup -Path XDHyp:\Connections\AWS -Name MySecurityGroup -Description 'Example group'
c:\PS> Grant-HypSecurityGroupEgress $Group.FullPath -Protocol '-1' -IPRange '0.0.0.0/16'
```

Create a security group and grant full egress to anywhere.

----- **EXAMPLE 2** -----

```
c:\PS> $Group1 = New-Item -ItemType SecurityGroup -Path XDHyp:\Connections\AWS -Name MySecurityGroup1 -Description 'Example group 1'
c:\PS> $Group2 = New-Item -ItemType SecurityGroup -Path XDHyp:\Connections\AWS\MySecurityGroup2 -Description 'Example group 2'
c:\PS> Grant-HypSecurityGroupEgress $Group1.FullPath -FromPort 0 -ToPort 0 -Protocol icmp -GroupId $Group2.Id
c:\PS> Grant-HypSecurityGroupIngress $Group2.FullPath -FromPort 0 -ToPort 0 -Protocol icmp -GroupId $Group1.Id
```

Make 2 security groups and permit group 1 to ping group 2.

# Grant-HypSecurityGroupIngress

Jan 04, 2017

Adds an ingress rule to a security group.

## Syntax

```
Grant-HypSecurityGroupIngress [-LiteralPath] <String> -GroupId <String[]> -Protocol <String> [-FromPort <Decimal>] [-ToPort <Decimal>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Grant-HypSecurityGroupIngress [-LiteralPath] <String> -IPRange <String[]> -Protocol <String> [-FromPort <Decimal>] [-ToPort <Decimal>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Adding an egress rule permits network traffic from source CIDR IP address ranges or security groups to pass to instances within a security group.

## Related topics

Amazon AuthorizeSecurityGroupEgress: <http://docs.aws.amazon.com/AWSEC2/latest/APIReference/ApiReference-query-AuthorizeSecurityGroupEgress.html>

IANA protocol numbers: <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>

[Grant-HypSecurityGroupEgress](#)

[Revoke-HypSecurityGroupIngress](#)

[Revoke-HypSecurityGroupEgress](#)

## Parameters

### -LiteralPath<String>

Specifies the full XD Hyp provider path to the security group, equivalent to the FullPath property of the security group object. The path can specify a security group relative to a hypervisor connection or hosting unit.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

### -Protocol<String>

Specifies the protocol name or number. Protocol numbers can be found at: <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>

Use -1 to specify all protocols.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-GroupId**<String[]>

Specifies one or more source security groups from which traffic will be permitted by this rule. This parameter cannot be specified in conjunction with IPRange.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-IPRange**<String[]>

Specifies one or more source CIDR IP address ranges from which traffic will be permitted by this rule. This parameter cannot be specified in conjunction with IPRange.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-FromPort**<Decimal>

The start of the port range for port based protocols. For ICMP this specifies the type number.

Use -1 to specify all ICMP types.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-ToPort**<Decimal>

The end of the port range for port based protocols. For ICMP this specifies the type number, where -1 can be used to specify all ICMP types.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and



Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Input Type**

System.string The LiteralPath can be piped in.

**Return Values**

None

**Notes**

Security groups can be added and removed using the New-Item and Remove-Item cmdlets.

**Examples**

----- **EXAMPLE 1** -----

```
c:\PS> $Group = New-Item -ItemType SecurityGroup -Path XDHyp:\Connections\AWS -Name MySecurityGroup -Description 'Example group'  
c:\PS> Grant-HypSecurityGroupIngress $Group.FullPath -FromPort 80 -ToPort 80 -Protocol tcp -IPRange '0.0.0.0/0'
```

Create a security group and grant ingress on port 80 from anywhere.

----- **EXAMPLE 2** -----

```
c:\PS> $Group1 = New-Item -ItemType SecurityGroup -Path XDHyp:\Connections\AWS -Name MySecurityGroup1 -Description 'Example group 1'  
c:\PS> $Group2 = New-Item -ItemType SecurityGroup -Path XDHyp:\Connections\AWS\MySecurityGroup2 -Description 'Example group 2'  
c:\PS> Grant-HypSecurityGroupEgress $Group1.FullPath -FromPort 8080 -ToPort 8080 -Protocol tcp -GroupId $Group2.Id  
c:\PS> Grant-HypSecurityGroupIngress $Group2.FullPath -FromPort 8080 -ToPort 8080 -Protocol tcp -GroupId $Group1.Id  
c:\PS> Grant-HypSecurityGroupEgress $Group2.FullPath -Protocol '-1' -GroupId $Group1.Id  
c:\PS> Grant-HypSecurityGroupIngress $Group1.FullPath -Protocol '-1' -GroupId $Group2.Id
```

Make 2 security groups and permit group 1 access to group 2 only on port 8080 while granting full access to group 1 from group 2.

# New-HypVMSnapshot

Jan 04, 2017

Creates a new snapshot for the specified VM item path.

## Syntax

```
New-HypVMSnapshot [-LiteralPath] <String> [-SnapshotName] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [[-SnapshotDescription] <String>] [-<CommonParameters>]
```

## Detailed Description

Use this command to create a new snapshot of a virtual machine, for a given Host Service provider path to a VM. The resulting snapshot can then be used for operations that require a snapshot to work.

## Related topics

## Parameters

### **-LiteralPath**<String>

Specifies the path within a hosting unit provider to the virtual machine item for which to create a new snapshot. The path specified must be in one of the following formats: <drive>:\Connections\<Connection Name>\<Item Path of VM object> or <drive>:\Connections\{<connection Uid>\<Item Path of VM object>} or <drive>:\HostingUnits\<HostingUnit Name>\<Item Path of VM object> or <drive>:\HostingUnits\{<hostingUnit Uid>\<Item Path of VM object>}

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

### **-SnapshotName**<String>

The name of the new snapshot. This is visible in the hypervisor management console.

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.

Accept Pipeline Input?	false
------------------------	-------

**-SnapshotDescription<String>**

The description to add to the snapshot. This is visible in the hypervisor management console.

Required?	false
Default Value	
Accept Pipeline Input?	false

**Input Type**

System.string You can pipe a string that contains a path to Get-HypConfigurationDataForItem

**Return Values**

System.string.

The provider path to the newly created snapshot.

**Notes**

In the case of failure, the following errors can result.

**Error Codes**

-----

**InputHypervisorItemPathInvalid**

The path provided is not to an item in a sub-directory of a connection item or a hosting unit item.

**InvalidHypervisorItemPath**

No item exists with the specified path.

**InvalidHypervisorItem**

The item specified by the path exists, but is not a VM Item.

**SnapshotNameAlreadyInUse**

The specified name is already in use and will cause a name resolution clash.

**FailedToCreateSnapshot**

The snapshot creation process failed.

**HypervisorInMaintenanceMode**

The hypervisor is in maintenance mode.

**DatabaseError**

An error occurred in the service while attempting a database operation.

**DatabaseNotConfigured**

The operation could not be completed because the database for the service is not configured.

**DataStoreException**

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

**CommunicationError**

An error occurred while communicating with the service.

PermissionDenied

The user does not have administrative rights to perform this operation.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

SnapshotChainTooLong

Snapshot creation failed. Snapshot chain is too long.

SnapshotCreationNotAuthorized

Snapshot creation failed. User not authorized to create snapshots.

Examples

----- **EXAMPLE 1** -----

```
C:\PS>New-HypVMSnapshot -LiteralPath XDHyp:\Connections\MyConnection\MyVm.vm -SnapshotName "New snapshot" -SnapshotDescription "Example snapshot"
```

```
    XDHyp:\Connections\MyConnection\MyVm.vm\New snapshot.snapshot
```

This command creates a snapshot of a VM called 'MyVm.vm' within a hypervisor connection called 'MyConnection'.

# Remove-HypHostingUnitMetadata

Jan 04, 2017

Removes metadata from the given HostingUnit.

## Syntax

```
Remove-HypHostingUnitMetadata [-HostingUnitUid] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHostingUnitMetadata [-HostingUnitUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHostingUnitMetadata [-HostingUnitName] <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHostingUnitMetadata [-HostingUnitName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHostingUnitMetadata [-InputObject] <HostingUnit[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHostingUnitMetadata [-InputObject] <HostingUnit[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given HostingUnit.

## Related topics

[Add-HypHostingUnitMetadata](#)

[Set-HypHostingUnitMetadata](#)

## Parameters

**-HostingUnitUid**<Guid>

Id of the HostingUnit

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-HostingUnitName**<String>

Name of the HostingUnit

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<HostingUnit[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create

high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-HypHostingUnit | % { Remove-HypHostingUnitMetadata -Map $_.MetadataMap }  
Remove all metadata from all HostingUnit objects.
```



# Remove-HypHostingUnitNetwork

Jan 04, 2017

Removes networks from a hosting unit.

## Syntax

```
Remove-HypHostingUnitNetwork [-LiteralPath] <String> [-NetworkPath] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this command to remove networks from a hosting unit. This does not remove the network from the hypervisor, only the reference to the network for the Host Service. After it is removed, the network is no longer available for associating with virtual NICs (when creating new virtual machines with the Machine Creation Service). Any virtual machines that have been created already continue to use the network until they are removed from the deployment. This command cannot be used if the connection for the hosting unit is in maintenance mode. If the network to be removed no longer exists on the hypervisor for the hosting unit, you must supply a fully qualified path to the network location.

## Related topics

[Add-HypHostingUnitNetwork](#)

## Parameters

### **-LiteralPath**<String>

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-NetworkPath**<String>

Specifies the path in the hosting unit provider of the network to remove. The path specified must be in one of the following formats: <drive>:\Connections\  
<HostingUnitName>\MyNetwork.network or <drive>:\Connections\{<hostingUnit Uid>\MyNetwork.network

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. This can be a host name or an IP address.

Required?	false
-----------	-------

Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Input Type

System.string You can pipe a string that contains a path to Remove-HypHostingUnitNetwork (Path parameter).

### Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### HostingUnitsPathInvalid

The path provided is not to an item in a subdirectory of a hosting unit item.

#### HostingUnitNetworkObjectToDeleteDoesNotExist

The network path specified is not part of the hosting unit.

#### HypervisorInMaintenanceMode

The hypervisor for the connection is in maintenance mode.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation. Communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used, or examine the XenDesktop logs.

### Examples

----- **EXAMPLE 1** -----

```
c:\PS>Remove-HypHostingUnitNetwork -LiteralPath XDHyp:\HostingUnits\MyHostingUnit -NetworkPath 'XDHyp:\HostingUnits\MyHostingUnits\newNetwork.network'
```

The command removes the network called "newNetwork.network" from the hosting unit called "MyHostingUnit"

# Remove-HypHostingUnitStorage

Jan 04, 2017

Removes storage from a hosting unit.

## Syntax

```
Remove-HypHostingUnitStorage [-LiteralPath] <String> [-StoragePath <String>] [-StorageType <StorageType>] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

## Detailed Description

Use this command to remove storage locations from a hosting unit. This does not remove the storage from the hypervisor, only the reference to the storage for the Host Service. After removal, the storage is no longer used to store hard disks (when creating new virtual machines with the Machine Creation Service). The hard disks located already on the storage remain in place and virtual machines that have been created already continue to use the storage until they are removed from the deployment. Do not use this command if the connection for the hosting unit is in maintenance mode. If the storage location to be removed no longer exists on the hypervisor for the hosting unit, you must supply a fully qualified path to the storage location.

## Related topics

[Add-HypHostingUnitStorage](#)

## Parameters

### **-LiteralPath**<String>

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-StoragePath**<String>

Specifies the path in the hosting unit provider of the storage to remove. If StoragePath is not specified, all storage is removed from the hosting unit. The path specified must be in one of the following formats: <drive>:\Connections\<HostingUnitName>\MyStorage.storage or <drive>:\Connections\{\<hostingUnit Uid>}\MyStorage.storage

Required?	false
Default Value	
Accept Pipeline Input?	true (ByValue)

### **-StorageType**<StorageType>

Specifies the type of storage in StoragePath. Supported storage types are: OSStorage PersonalDiskStorage

Required?	false
Default Value	OSStorage
Accept Pipeline Input?	false

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. This can be a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Input Type**

System.String You can pipe a string that contains a path to Remove-HypHostingUnitStorage (Path parameter).

**Notes**

After storage is removed, it is the administrator's responsibility to maintain its contents. The Citrix XenDesktop Machine Creation Service does not attempt to clean up any data that is stored in the storage location.

If all storage is removed from the hosting unit, other features of the Machine Creation Service stops functioning until some storage is added again.

In the case of failure, the following errors can result.

**Error Codes**

-----

**HostingUnitsPathInvalid**

The path provided is not to an item in a subdirectory of a hosting unit item.

**HostingUnitStorageObjectToDeleteDoesNotExist**

The storage path specified is not part of the hosting unit.

**HypervisorInMaintenanceMode**

The hypervisor for the connection is in maintenance mode.

**DatabaseError**

An error occurred in the service while attempting a database operation.

**DatabaseNotConfigured**

The operation could not be completed because the database for the service is not configured.

**DataStoreException**

An error occurred in the service while attempting a database operation. Communication with the database failed for various reasons.

**CommunicationError**

An error occurred while communicating with the service.

**PermissionDenied**

The user does not have administrative rights to perform this operation.

## ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used, or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Remove-HypHostingUnitStorage -LiteralPath XDHyp:\HostingUnits\MyHostingUnit -StoragePath 'XDHyp:\HostingUnits\MyHostingUnits\newStorage.storage'
```

The command removes the OS storage location called "newStorage.storage" from the hosting unit called "MyHostingUnit"

### ----- EXAMPLE 2 -----

```
c:\PS>Get-ChildItem XDHYP:\HostingUnits\Host\*.storage | Remove-HypHostingUnitStorage XDHYP:\HostingUnits\Host1
```

The command removes all OS storage from all hosting units called "Host1".

### ----- EXAMPLE 3 -----

```
c:\PS>Get-ChildItem XDHYP:\HostingUnits\Host\*.storage | Remove-HypHostingUnitStorage -StorageType PersonalVDiskStorage
```

The command removes all PersonalVDisk storage from all hosting units called "Host1".

# Remove-HypHypervisorConnectionAddress

Jan 04, 2017

Removes addresses from the list of available connection addresses.

## Syntax

```
Remove-HypHypervisorConnectionAddress [-LiteralPath] <String> [-HypervisorAddress] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this command to remove addresses that can be used to connect to the hypervisor specified by the hypervisor connection. If all addresses are removed, the connection cannot be used until a valid address is added to the hypervisor connection.

## Related topics

[Add-HypHypervisorConnectionAddress](#)

## Parameters

### **-LiteralPath**<String>

Specifies the path within a Host Service provider to the hosting unit item to which to add the address. The path specified must be in one of the following formats: <drive>:\HostingUnits\<HostingUnitName> or <drive>:\HostingUnits\{HostingUnit Id}>

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-HypervisorAddress**<String>

Specifies the address to be removed. If this parameter is not provided, all addresses are removed from the hypervisor connection.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Input Type

Systemstring You can pipe a string that contains a path to Remove-HypHypervisorConnectionAddress (Path parameter).

### Notes

Changes to a hypervisor connection affect all entities that reference the connection. If all addresses are removed from the connection, any other entities that reference the hypervisor connection (e.g. hosting units) cannot make new connections to the hypervisor.

In the case of failure, the following errors can result.

### Error Codes

-----

#### InputConnectionsPathInvalid

The path provided is not to an item in a subdirectory of a hosting unit item.

#### HypervisorConnectionAddressForeignKeyObjectDoesNotExist

The hypervisor connection to which the address is to be added could not be located.

#### HypervisorInMaintenanceMode

The hypervisor for the connection is in maintenance mode.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

### Examples

----- **EXAMPLE 1** -----

```
c:\PS>Remove-HypHypervisorConnectionAddress -LiteralPath XDHyp:\HostingUnits\MyHypervisorConnection -HypervisorAddress 'http:\myserver.com'
```

The command removes the address "http:\myserver.com" from the hypervisor connection called "MyHypervisorConnection".

----- **EXAMPLE 2** -----

```
c:\PS>Get-Item -Path XDHYP:\Connections\* | Remove-HypHypervisorConnectionAddress
```

The command removes all addresses from all the hypervisor connections currently defined.



# Remove-HypHypervisorConnectionMetadata

Jan 04, 2017

Removes metadata from the given HypervisorConnection.

## Syntax

```
Remove-HypHypervisorConnectionMetadata [-HypervisorConnectionUid] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHypervisorConnectionMetadata [-HypervisorConnectionUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given HypervisorConnection.

## Related topics

[Add-HypHypervisorConnectionMetadata](#)

[Set-HypHypervisorConnectionMetadata](#)

## Parameters

**-HypervisorConnectionUid**<Guid>

Id of the HypervisorConnection

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-HypervisorConnectionName**<String>

Name of the HypervisorConnection

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<HypervisorConnection[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create

high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-HypHypervisorConnection | % { Remove-HypHypervisorConnectionMetadata -Map $_.MetadataMap }  
Remove all metadata from all HypervisorConnection objects.
```

# Remove-HypHypervisorConnectionScope

Jan 04, 2017

Remove the specified HypervisorConnection(s) from the given scope(s).

## Syntax

```
Remove-HypHypervisorConnectionScope [-Scope] <String[]> -InputObject <HypervisorConnection[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHypervisorConnectionScope [-Scope] <String[]> -HypervisorConnectionUid <Guid[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHypervisorConnectionScope [-Scope] <String[]> -HypervisorConnectionName <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The RemoveHypHypervisorConnectionScope cmdlet is used to remove one or more HypervisorConnection objects from the given scope(s).

There are multiple parameter sets for this cmdlet, allowing you to identify the HypervisorConnection objects in different ways:

- HypervisorConnection objects can be piped in or specified by the InputObject parameter
- The HypervisorConnectionUid parameter specifies objects by HypervisorConnectionUid
- The HypervisorConnectionName parameter specifies objects by HypervisorConnectionName (supports wildcards)

To remove a HypervisorConnection from a scope you need permission to change the scopes of the HypervisorConnection.

If the HypervisorConnection is not in a specified scope, that scope will be silently ignored.

## Related topics

[Add-HypHypervisorConnectionScope](#)

[Get-HypScopedObject](#)

## Parameters

**-Scope**<String[]>

Specifies the scopes to remove the objects from.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-InputObject**<HypervisorConnection[]>

Specifies the HypervisorConnection objects to be removed.

Required?	true
Default Value	

Accept Pipeline Input?	true (ByValue, ByPropertyName)
------------------------	--------------------------------

**-HypervisorConnectionUid**<Guid[]>

Specifies the HypervisorConnection objects to be removed by HypervisorConnectionUid.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-HypervisorConnectionName**<String[]>

Specifies the HypervisorConnection objects to be removed by HypervisorConnectionName.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

None

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### ScopeNotFound

One of the specified scopes was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command with the specified objects or scopes.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Remove-HypHypervisorConnectionScope Finance -HypervisorConnectionUid 6702C5D0-C073-4080-A0EE-EC74CB537C52  
Removes a single HypervisorConnection from the 'Finance' scope.
```

----- **EXAMPLE 2** -----

```
c:\PS>Remove-HypHypervisorConnectionScope Finance,Marketing -HypervisorConnectionUid 6702C5D0-C073-4080-A0EE-EC74CB537C52  
Removes a single HypervisorConnection from multiple scopes.
```

----- **EXAMPLE 3** -----

```
c:\PS>Get-HypHypervisorConnection | Remove-HypHypervisorConnectionScope Finance
```

Removes all visible HypervisorConnection objects from the 'Finance' scope.

----- **EXAMPLE 4** -----

```
c:\PS>Remove-HypHypervisorConnectionScope Finance -HypervisorConnectionName A*
```

Removes HypervisorConnection objects with a name starting with an 'A' from the 'Finance' scope.



# Remove-HypMetadata

Jan 04, 2017

Removes metadata from a hypervisor connection or hosting unit.

## Syntax

```
Remove-HypMetadata [-LiteralPath] <String> [-Property <String>] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Use this command to remove custom data from a specific hosting unit or hypervisor connection. If the Property parameter is not provided, all metadata is removed from the specified item.

## Related topics

[Remove-HypMetadata](#)

## Parameters

### **-LiteralPath**<String>

Specifies the path within a Host Service provider to the hosting unit or hypervisor connection item from which to remove the metadata. The path specified must be in one of the following formats: <drive>:\HostingUnits\<HostingUnitName> or <drive>:\HostingUnits\{<HostingUnit Uid> or <drive>:\Connections\<Connection Name> or <drive>:\Connections\{<Connection Uid>

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

### **-Property**<String>

Specifies the property name of the metadata to be removed.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

#### Input Type

Systemstring You can pipe a string that contains a path to Add-HypMetadata (Path parameter).

#### Notes

In the case of failure, the following errors can result.

#### Error Codes

-----

#### InvalidPath

The path provided is not in the required format.

#### HostingUnitMetadataObjectToDeleteDoesNotExist

The hosting unit supplied in the path does not exist.

#### HypervisorConnectionObjectToDeleteDoesNotExist

The hypervisor connection supplied in the path does not exist.

#### MetadataContainerUndefined

The specified path does not reference a hosting unit or a hypervisor connection.

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## CommunicationError

An error occurred while communicating with the service.

## PermissionDenied

The user does not have administrative rights to perform this operation.

## ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS>Remove-HypMetadata -LiteralPath XDHyp:\Connections\MyConnection -Property MyProperty
```

The command removes the metadata with the property "MyProperty" from the hypervisor connection called "MyConnection".

### ----- EXAMPLE 2 -----

```
C:\PS>Remove-HypMetadata -LiteralPath XDHyp:\Connections\MyConnection
```

The command removes all the metadata from the hypervisor connection called "MyConnection".

### ----- EXAMPLE 3 -----

```
C:\PS>dir XDHyp:\connections | Remove-HypMetadata
```

The command removes all the metadata from all the hypervisor connections.

# Remove-HypServiceMetadata

Jan 04, 2017

Removes metadata from the given Service.

## Syntax

```
Remove-HypServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Service.

## Related topics

[Set-HypServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<Service[]>

Objects to which metadata is to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Map<PSObject>**

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name<String>**

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId<Guid>**

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
-----------	-------

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-HypService | % { Remove-HypServiceMetadata -Map $_.MetadataMap }  
Remove all metadata from all Service objects.
```

# Reset-HypServiceGroupMembership

Jan 04, 2017

Reloads the access permissions and configuration service locations for the Host Service.

## Syntax

```
Reset-HypServiceGroupMembership [-ConfigServiceInstance] <ServiceInstance[]> [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

## Detailed Description

Enables you to reload Host Service access permissions and configuration service locations. The `Reset-HypServiceGroupMembership` command must be run on at least one instance of the service type (Hyp) after installation and registration with the configuration service. Without this operation, the Host services will be unable to communicate with other services in the XenDesktop deployment. When the command is run, the services are updated when additional services are added to the deployment, provided that the configuration service is not stopped. The `Reset-HypServiceGroupMembership` command can be run again to refresh this information if automatic updates do not occur when new services are added to the deployment. If more than one configuration service instance is passed to the command, the first instance that meets the expected service type requirements is used.

## Related topics

### Parameters

**-ConfigServiceInstance**<ServiceInstance[]>

Specifies the configuration service instance object that represents the service instance for the type 'InterService' that references a configuration service for the deployment.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the `Start-LogHighLevelOperation` and `Stop-LogHighLevelOperation` cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
-----------	-------



Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Input Type

Citrix.Host.Sdk.ServiceInstance[] Service instances containing a ServiceInstance object that refers to the central configuration service interservice interface can be piped to the Reset-HypServiceGroupMembership command.

### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### NoSuitableServiceInstance

None of the supplied service instance objects were suitable for resetting service group membership.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

### Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-HypServiceGroupMembership
```

Reset the service group membership for a service in a deployment where the configuration service is configured and running on the same machine as the service.

----- **EXAMPLE 2** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config -AdminAddress OtherServer.example.com | Reset-HypServiceGroupmembership
```

Reset the service group membership for a service in a deployment where the configuration service that is configured and running on a machine named 'OtherServer.example.com'.

# Revoke-HypSecurityGroupEgress

Jan 04, 2017

Removes an egress rule from a security group.

## Syntax

```
Revoke-HypSecurityGroupEgress [-LiteralPath] <String> -GroupId <String[]> -Protocol <String> [-FromPort <Decimal>] [-ToPort <Decimal>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Revoke-HypSecurityGroupEgress [-LiteralPath] <String> -IPRange <String[]> -Protocol <String> [-FromPort <Decimal>] [-ToPort <Decimal>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

To remove a rule, specify parameters matching an existing rule's values.

## Related topics

Amazon AuthorizeSecurityGroupEgress: <http://docs.aws.amazon.com/AWSEC2/latest/APIReference/ApiReference-query-AuthorizeSecurityGroupEgress.html>

IANA protocol numbers: <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>

[Grant-HypSecurityGroupIngress](#)

[Grant-HypSecurityGroupEgress](#)

[Revoke-HypSecurityGroupIngress](#)

## Parameters

### **-LiteralPath**<String>

Specifies the full XD Hyp provider path to the security group, equivalent to the FullPath property of the security group object. The path can specify a security group relative to a hypervisor connection or hosting unit.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

### **-Protocol**<String>

Specifies the protocol name or number. Protocol numbers can be found at: <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>

Use -1 to specify all protocols.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-GroupId**<String[]>

Specifies one or more destination security groups to which traffic will be permitted by this rule. This parameter cannot be specified in conjunction with IPRange.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-IPRange**<String[]>

Specifies one or more destination CIDR IP address ranges to which traffic will be permitted by this rule. This parameter cannot be specified in conjunction with IPRange.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-FromPort**<Decimal>

The start of the port range for port based protocols. For ICMP this specifies the type number.

Use -1 to specify all ICMP types.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-ToPort**<Decimal>

The end of the port range for port based protocols. For ICMP this specifies the type number, where -1 can be used to specify all ICMP types.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Input Type

System.string The LiteralPath can be piped in.

Return Values

None

Notes

Security groups cannot be removed in AWS if they are referened by rules from other security groups.

Security groups can be added and removed using the New-Item and Remove-Item cmdlets.

Examples

----- **EXAMPLE 1** -----

```
c:\PS> $Group = New-Item -ItemType SecurityGroup -Path XDHyp:\Connections\AWS -Name MySecurityGroup -Description 'Example group'
c:\PS> Grant-HypSecurityGroupEgress $Group.FullPath -Protocol '-1' -IPRange '0.0.0.0/0'
c:\PS> Revoke-HypSecurityGroupEgress $Group.FullPath -Protocol '-1' -IPRange '0.0.0.0/0'
c:\PS> Remove-Item $Group.FullPath
```

Create a security group, grant full egress to anywhere, then revoke access and delete the security group.

# Revoke-HypSecurityGroupIngress

Jan 04, 2017

Removes an ingress rule from a security group.

## Syntax

```
Revoke-HypSecurityGroupIngress [-LiteralPath <String> -GroupId <String[]> -Protocol <String> [-FromPort <Decimal>] [-ToPort <Decimal>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Revoke-HypSecurityGroupIngress [-LiteralPath <String> -IPRange <String[]> -Protocol <String> [-FromPort <Decimal>] [-ToPort <Decimal>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

To remove a rule, specify parameters matching an existing rule's values.

## Related topics

Amazon AuthorizeSecurityGroupEgress: <http://docs.aws.amazon.com/AWSEC2/latest/APIReference/ApiReference-query-AuthorizeSecurityGroupEgress.html>

IANA protocol numbers: <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>

[Grant-HypSecurityGroupIngress](#)

[Grant-HypSecurityGroupEgress](#)

[Revoke-HypSecurityGroupIngress](#)

## Parameters

**-LiteralPath**<String>

Specifies the full XD Hyp provider path to the security group, equivalent to the FullPath property of the security group object. The path can specify a security group relative to a hypervisor connection or hosting unit.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Protocol**<String>

Specifies the protocol name or number. Protocol numbers can be found at: <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>

Use -1 to specify all protocols.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-GroupId**<String[]>

Specifies one or more source security groups from which traffic will be permitted by this rule. This parameter cannot be specified in conjunction with IPRange.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-IPRange**<String[]>

Specifies one or more source CIDR IP address ranges from which traffic will be permitted by this rule. This parameter cannot be specified in conjunction with IPRange.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-FromPort**<Decimal>

The start of the port range for port based protocols. For ICMP this specifies the type number.

Use -1 to specify all ICMP types.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-ToPort**<Decimal>

The end of the port range for port based protocols. For ICMP this specifies the type number, where -1 can be used to specify all ICMP types.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	

Accept Pipeline Input?	false
------------------------	-------

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Input Type**

Systemstring The LiteralPath can be piped in.

**Return Values**

None

**Notes**

Security groups cannot be removed in AWS if they are referened by rules from other security groups.

Security groups can be added and removed using the New-Item and Remove-Item cmdlets.

**Examples**

----- **EXAMPLE 1** -----

```
c:\PS> $Group1 = New-Item -ItemType SecurityGroup -Path XDHyp:\Connections\AWS -Name MySecurityGroup1 -Description 'Example group 1'
c:\PS> $Group2 = New-Item -ItemType SecurityGroup -Path XDHyp:\Connections\AWS -Name MySecurityGroup2 -Description 'Example group 2'
c:\PS> Grant-HypSecurityGroupEgress $Group1.FullPath -FromPort 8080 -ToPort 8085 -Protocol tcp -GroupId $Group2.Id
c:\PS> Grant-HypSecurityGroupIngress $Group2.FullPath -FromPort 8080 -ToPort 8085 -Protocol tcp -GroupId $Group1.Id
c:\PS> Revoke-HypSecurityGroupEgress $Group1.FullPath -FromPort 8080 -ToPort 8085 -Protocol tcp -GroupId $Group2.Id
c:\PS> Revoke-HypSecurityGroupIngress $Group2.FullPath -FromPort 8080 -ToPort 8085 -Protocol tcp -GroupId $Group1.Id
```

Create 2 security groups, grant access from group 1 to group 2, then revoke access.



# Set-HypAdminConnection

Jan 04, 2017

Set the controller to be used by the cmdlets that form the Host service PowerShell snap-in.

## Syntax

```
Set-HypAdminConnection [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this command to set the default controller address to be used by the cmdlets to communicate with the controller. Most Host service cmdlets take an 'AdminAddress' parameter that can be used to define the controller (when used, this overrides this setting). However, the Set-Location cmdlet in the Host service provider does not offer this option. Therefore, the controller address must be set using this cmdlet, if no other cmdlets have set the address previously in the current runspace.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Set-HypAdminConnection -AdminAddress myserver.com
```

This command sets the controller address for the commands to be "myserver.com". All commands run use this address until it is altered, either by another call to this command or by the use of the 'AdminAddress' parameter in another command in the Host service snap-in.

# Set-HypDBConnection

Jan 04, 2017

Configures a database connection for the Host Service.

## Syntax

```
Set-HypDBConnection [-DBConnection] <String> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Configures a connection to a database in which the Host Service can store its state. The service will attempt to connect and start using the database immediately after the connection is configured. The database connection string is updated to the specified value regardless of whether it is valid or not. Specifying an invalid connection string prevents a service from functioning until the error is corrected.

After a connection is configured, you cannot alter it without first clearing it (by setting the connection to \$null).

You do not need to configure a database connection to use this command.

## Related topics

[Get-HypServiceStatus](#)

[Get-HypDBConnection](#)

[Test-HypDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be used by the Host Service. Passing in \$null will clear any existing database connection configured.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Force**<SwitchParameter>

If present, allows the local administrator to set the connection string to null when there are problems contacting the database or other services.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and

Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Set-HypDBConnection command returns an object containing the status of the Host Service together with extra diagnostics information.

DBUnconfigured

The Host Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the Host Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Host Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the Host Service currently in use is incompatible with the version of the Host Service schema on the database. Upgrade the Host Service to a more recent version.

DBOlderVersionThanService

The version of the Host Service schema on the database is incompatible with the version of the Host Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The Host Service is running and is connected to a database containing a valid schema.

Failed

The Host Service has failed.

Unknown

The status of the Host Service cannot be determined.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidDBConnectionString

The database connection string has an invalid format.

DatabaseConnectionDetailsAlreadyConfigured

There was already a database connection configured. After a configuration is set, it can only be set to \$null.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-HypDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Configures a database connection string for the Host Service.

----- **EXAMPLE 2** -----

```
c:\PS>Set-HypDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Configures an invalid database connection string for the Host Service.

# Set-HypHostingUnitMetadata

Jan 04, 2017

Adds or updates metadata on the given HostingUnit.

## Syntax

```
Set-HypHostingUnitMetadata [-HostingUnitUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypHostingUnitMetadata [-HostingUnitUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypHostingUnitMetadata [-HostingUnitName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypHostingUnitMetadata [-HostingUnitName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypHostingUnitMetadata [-InputObject] <HostingUnit[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypHostingUnitMetadata [-InputObject] <HostingUnit[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given HostingUnit objects.

## Related topics

[Add-HypHostingUnitMetadata](#)

[Remove-HypHostingUnitMetadata](#)

## Parameters

**-HostingUnitUid**<Guid>

Id of the HostingUnit

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-HostingUnitName**<String>

Name of the HostingUnit

Required?	true
-----------	------

Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<HostingUnit[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the HostingUnit specified. The property cannot contain any of the following characters \/:;#. \*?=<> | []()"

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Value**<String>

Specifies the value for the property.

Required?	true

Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Collections.Generic.Dictionary[String,String]

Set-HypHostingUnitMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidParameterCombination



The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-HypHostingUnitMetadata -HostingUnitUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	-----
property	value

Add metadata with a name of 'property' and a value of 'value' to the HostingUnit with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Set-HypHostingUnitStorage

Jan 04, 2017

Sets options for a storage location on a hosting unit.

## Syntax

```
Set-HypHostingUnitStorage [-LiteralPath] <String> [-StoragePath] <String> [-StorageType <StorageType>] [-Superseded <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this command to set options for storage locations that are defined for a hosting unit. Do not use this command if the connection for the hosting unit is in maintenance mode.

## Related topics

## Parameters

### **-LiteralPath**<String>

Specifies the path to the hosting unit which defines the storage. The path must be in one of the following formats: <drive>:\HostingUnits\<HostingUnitName> or <drive>:\HostingUnits\{<HostingUnit Uid>}

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-StoragePath**<String>

Specifies the path to the storage that will be modified. The path must be in one of the following formats: <drive>:\Connections\<ConnectionName>\MyStorage.storage or <drive>:\Connections\{<Connection Uid>\MyStorage.storage or <drive>:\HostingUnits\<HostingUnitName>\MyStorage.storage or <drive>:\HostingUnits\{<hostingUnit Uid>\MyStorage.storage

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

### **-StorageType**<StorageType>

Specifies the type of storage in StoragePath. Supported storage types are: OSStorage PersonalDiskStorage

Required?	false
Default Value	OSStorage
Accept Pipeline Input?	false

### **-Superseded**<Boolean>

Specifies that this storage has been superseded and must not be used for future provisioning operations. Existing virtual machines using this storage will continue to function.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects. This can be a host name or an IP address.

Required?	false
Default Value	LocalHost. When a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Input Type**

Systemstring You can pipe a string that contains a path to Add-HypHostingUnitStorage (StoragePath parameter).

**Return Values**

Citrix.XDPowerShell.HostingUnit

Add-HypHostingUnitStorage returns an object containing the new definition of the hosting unit.

HostingUnitUid <Guid>

Specifies the unique identifier for the hosting unit.

HostingUnitName <string>

Specifies the name of the hosting unit.

HypervisorConnection <Citrix.XDPowerShell.HypervisorConnection>

Specifies the connection that the hosting unit uses to access a hypervisor.

RootId <string>

Identifies, to the hypervisor, the root of the hosting unit.

RootPath <string>

The hosting unit provider path that represents the root of the hosting unit.

Storage <Citrix.XDPowerShell.Storage[]>

The list of storage items that the hosting unit can use.

PersonalVDiskStorage <Citrix.XDPowerShell.Storage[]>

The list of storage items that the hosting unit can use for storing personal data.

VMTaggingEnabled <Boolean>

Specifies whether or not the metadata in the hypervisor can be used to store information about the XenDesktop Machine Creation Service.

NetworkId <string>

The hypervisor's internal identifier that represents the network specified for the hosting unit.

NetworkPath <string>

The hosting unit provider path to the network specified for the hosting unit.

Metadata <Citrix.XDPowerShell.Metadata[]>

A list of key value pairs that can store additional information about the hosting unit.

## Notes

The storage path must be valid for the hosting unit. The rules that are applied are as follows: XenServer (HypervisorConnection Type = XenServer)

NA

VMWare vSphere/ESX (HypervisorConnection Type = vCenter)

The storage path must be directly contained in the root path item of the hosting unit.

Microsoft SCVMM/Hyper-v (HypervisorConnection Type = SCVMM)

Only one storage entry for these connection types is valid, and it must reference an SMB share. Additionally, if a Hyper-V failover cluster is used the SMB share must be the top-level mount point of the cluster shared volume on one of the servers in the cluster (i.e. C:\ClusterStorage).

In the case of failure, the following errors can be produced.

Error Codes

-----

HostingUnitsPathInvalid

The path provided is not to an item in a subdirectory of a hosting unit item.

HostingUnitStoragePathInvalid

The specified path is invalid.

HostingUnitStoragePathInvalid

The storage path cannot be found or is invalid. See notes above about validity.

HostingUnitStorageDuplicateObjectExists

The specified storage path is already part of the hosting unit.

HypervisorInMaintenanceMode

The hypervisor for the connection is in maintenance mode.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNot Configured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation. Communication with the database failed for various reasons.

CommunicationError

An error occurred while communicating with the service.

PermissionDenied

The user does not have administrative rights to perform this operation.

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used, or examine the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-HypHostingUnitStorage -LiteralPath XDHyp:\HostingUnits\MyHostingUnit -StoragePath 'XDHyp:\HostingUnits\MyHostingUnits\newStorage.storage' -Superseded $true
```

```
HostingUnitUid      : bcd3d649-86d1-4aa8-8ec2-d322b6a2c457
HostingUnitName     : MyHostingUnit
HypervisorConnection : MyConnection
RootPath            : /
RootId              :
NetworkPath         : /Network 0.network
```

```
NetworkId      : ab47080b-ca15-771a-c8dc-6ad9650158f1
Storage        : {/Local storage.storage, /newStorage.storage}
PersonalvDiskStorage : {/newStorage.storage}
VMTaggingEnabled : True
Metadata       : {}
```

The command updates a storage location called "newStorage.storage" associated with the hosting unit called "MyHostingUnit".

# Set-HypHypervisorConnectionMetadata

Jan 04, 2017

Adds or updates metadata on the given HypervisorConnection.

## Syntax

```
Set-HypHypervisorConnectionMetadata [-HypervisorConnectionUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypHypervisorConnectionMetadata [-HypervisorConnectionUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given HypervisorConnection objects.

## Related topics

[Add-HypHypervisorConnectionMetadata](#)

[Remove-HypHypervisorConnectionMetadata](#)

## Parameters

**-HypervisorConnectionUid**<Guid>

Id of the HypervisorConnection

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-HypervisorConnectionName**<String>

Name of the HypervisorConnection

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<HypervisorConnection[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the HypervisorConnection specified. The property cannot contain any of the following characters \/:#. \*?=<> | []()''

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Value**<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

System.Collections.Generic.Dictionary[String,String]

Set-HypHypervisorConnectionMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

##### InvalidParameterCombination

The cmdlet parameters are inconsistent.

##### UnknownObject

One of the specified objects was not found.

##### DatabaseError

An error occurred in the service while attempting a database operation.

##### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

##### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

##### PermissionDenied

You do not have permission to execute this command.

##### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

##### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.



## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Set-HypHypervisorConnectionMetadata -HypervisorConnectionUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	----
property	value

Add metadata with a name of 'property' and a value of 'value' to the HypervisorConnection with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Set-HypServiceMetadata

Jan 04, 2017

Adds or updates metadata on the given Service.

## Syntax

```
Set-HypServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-HypServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Allows you to store additional custom data against given Service objects.

## Related topics

[Remove-HypServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<Service[]>

Objects to which metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Service specified. The property cannot contain any of the following characters `\;#.*?=<>|[]()`

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Value**<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with `@{"name1" = "val1"; "name2" = "val2"}`) or a string dictionary (created with `new-object "System.Collections.Generic.Dictionary[String,String]"`).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the `Start-LogHighLevelOperation` and `Stop-LogHighLevelOperation` cmdlets.

Required?	false
-----------	-------

Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

System.Collections.Generic.Dictionary[String,String]

Set-HypServiceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

**Notes**

If the command fails, the following errors can be returned.

**Error Codes**

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

### Examples

#### ----- EXAMPLE 1 -----

```
c:\PS>Set-HypServiceMetadata -ServiceHostId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Service with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Set-HypVolumeServiceConfiguration

Jan 04, 2017

Applies a change to one of the VolumeServiceConfiguration instances in the site.

## Syntax

```
Set-HypVolumeServiceConfiguration -VolumeWorkerPackageUri <String> -ConnectionType <String> -VolumeServiceConfigurationName <String> [-LoggingId <Guid>] [-AdminAddress <String>] [-CommonParameters<>]
```

```
Set-HypVolumeServiceConfiguration -RegionName <String> -BaseLinuxTemplateId <String> -ConnectionType <String> -VolumeServiceConfigurationName <String> [-LoggingId <Guid>] [-AdminAddress <String>] [-CommonParameters<>]
```

## Detailed Description

Volume service configurations are used to control how cloud-based host connections behave when provisioning machines. They contain two pieces of information. The first is a per-region specification of the cloud template that provides the standard Linux operating system for the cloud. The second is the specification of a URL from which the Citrix Volume Worker software can be installed (not all cloud connections make use of this URL, but they all make use of the template map).

Each cmdlet invocation can be used to either change the volume worker URL, or to modify (or add) an entry in the Linux template map. These two operations are supported by parameter sets. To change both properties, you must invoke the cmdlet twice.

## Related topics

[Get-HypVolumeServiceConfiguration](#)

## Parameters

### **-ConnectionType**<String>

Specifies the cloud connection type, such as "AWS" or "CloudPlatform".

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-VolumeServiceConfigurationName**<String>

Specifies the name of the configuration you want to modify. This parameter is used alongside the ConnectionType to specify a single configuration set unambiguously. There can be only one named configuration per connection type. In a newly-configured site, there will be exactly one configuration set called "SiteDefault" for each of CloudPlatform and AWS.

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-VolumeWorkerPackageUri**<String>

Specifies a (new) URI for the volume worker package.

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-RegionName**<String>

Specifies the cloud region in which the Linux template resides. This parameter is used only when passing the BaseLinuxTemplateId parameter. The format of the string is cloud-specific. For example, for an AWS-based cloud, it would be a region identifier such as "us-east-1".

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-BaseLinuxTemplateId**<String>

Specifies a change to the standard Linux template that should be used in the cloud. When passing this parameter, you must also specify the RegionName parameter to indicate the region in which

this template resides. The format of the string is cloud-specific. For example, for an AWS-based cloud, it would be an AMI identifier such as "ami-1234abcd".

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

Citrix.Host.Sdk.VolumeServiceConfiguration

This cmdlet returns the updated VolumeServiceConfiguration object.

**Examples**

----- **EXAMPLE 1** -----

C:\PS>Set-HypVolumeServiceConfiguration -VolumeServiceConfigurationName SiteDefault -ConnectionType CloudPlatform -RegionName Region1 -BaseLinuxTemplateId 9883ba7a-74bf-4002-9b12-1  
This command specifies a Linux template that should be used by default for CloudPlatform connections.

----- **EXAMPLE 2** -----

C:\PS>Set-HypVolumeServiceConfiguration -VolumeServiceConfigurationName SiteDefault -ConnectionType CloudPlatform -VolumeWorkerPackageUri "http://cloudadmin.net/citrix/volumeworker/versic  
This command specifies a worker package download URL that should be used by default for CloudPlatform connections.

# Start-HypVM

Jan 04, 2017  
Starts a VM.

## Syntax

```
Start-HypVM [-LiteralPath] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

This command provides a mechanism to start a VM.

## Related topics

### Parameters

#### **-LiteralPath**<String>

Specifies the path within a hosting unit provider to the virtual machine item to start. The path specified must be in one of the following formats: <drive>:\Connections\<Connection Name>\<Item Path of VM object> or <drive>:\Connections\{<connection Uid>\<Item Path of VM object>} or <drive>:\HostingUnits\<HostingUnit Name>\<Item Path of VM object> or <drive>:\HostingUnits\{<hostingUnit Uid>\<Item Path of VM object>}

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Input Type

System.String You can pipe a string that contains a path.

## Return Values



System.void.

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### InputHypervisorItemPathInvalid

The path provided is not to an item in a sub-directory of a connection item or a hosting unit item.

#### InvalidHypervisorItemPath

No item exists with the specified path.

#### InvalidHypervisorItem

The item specified by the path exists, but is not a VM Item.

#### HypervisorInMaintenanceMode

The hypervisor is in maintenance mode.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### CommunicationError

An error occurred while communicating with the service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS>Start-HypVM -LiteralPath XDHyp:\Connections\MyConnection\MyVm.vm
```

This command starts a VM called 'MyVm.vm' within a hypervisor connection called 'MyConnection'.

# Stop-HypVM

Jan 04, 2017

Stops a VM by issuing a Shutdown request

## Syntax

```
Stop-HypVM [-LiteralPath] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this command to change the power state of a VM from running to stopped.

## Related topics

### Parameters

#### **-LiteralPath**<String>

Specifies the path within a hosting unit provider to the virtual machine item to stop. The path specified must be in one of the following formats: <drive>:\Connections\<Connection Name>\<Item Path of VM object> or <drive>:\Connections\{<connection Uid>\<Item Path of VM object>} or <drive>:\HostingUnits\<HostingUnit Name>\<Item Path of VM object> or <drive>:\HostingUnits\{<hostingUnit Uid>\<Item Path of VM object>}

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in will connect to. This can be provided as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

## Input Type

System.String You can pipe a string that contains a path.

## Return Values

System.void.

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### InputHypervisorItemPathInvalid

The path provided is not to an item in a sub-directory of a connection item or a hosting unit item.

#### InvalidHypervisorItemPath

No item exists with the specified path.

#### InvalidHypervisorItem

The item specified by the path exists, but is not a VM Item.

#### HypervisorInMaintenanceMode

The hypervisor is in maintenance mode.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### CommunicationError

An error occurred while communicating with the service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS>Stop-HypVM -LiteralPath XDHyp:\Connections\MyConnection\MyVm.vm
```

This command stops a VM called 'MyVm.vm' within a hypervisor connection called 'MyConnection'.

# Test-HypDBConnection

Jan 04, 2017

Tests a database connection for the Host Service.

## Syntax

```
Test-HypDBConnection [-DBConnection] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Tests a connection to the database in which the Host Service can store its state. The service will attempt to connect to the database without affecting the current connection to the database.

You do not have to clear the connection to use this command.

## Related topics

[Get-HypServiceStatus](#)

[Get-HypDBConnection](#)

[Set-HypDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be tested by the Host Service.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
-----------	-------

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Test-HypDBConnection command returns an object containing the status of the Host Service if the connection string of the specified data store were to be set to the string being tested, together with extra diagnostics information for the specified connection string.

DBRejectedConnection

The database rejected the logon attempt from the Host Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Host Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the Host Service currently in use is incompatible with the version of the Host Service schema on the database. Upgrade the Host Service to a more recent version.

DBOlderVersionThanService

The version of the Host Service schema on the database is incompatible with the version of the Host Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The Set-HypDBConnection command would succeed if it were executed with the supplied connection string.

Failed

The Host Service has failed.

Unknown

The status of the Host Service cannot be determined.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

InvalidDBConnectionString

The database connection string has an invalid format.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Test-HypDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Tests a database connection string for the Host Service.

----- **EXAMPLE 2** -----

```
c:\PS>Test-HypDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Tests an invalid database connection string for the Host Service.

# Test-HypHostingUnitNameAvailable

Jan 04, 2017

Checks to ensure that the proposed name for a hosting unit is unused.

## Syntax

```
Test-HypHostingUnitNameAvailable -HostingUnitName <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Use this command to check that the proposed name for a hosting unit is unused. This check is done without regard for scoping of the existing hosting unit, so the names of inaccessible hosting units are also checked.

## Related topics

[New-Item](#)

[Rename-Item](#)

## Parameters

**-HostingUnitName**<String[]>

The name or names of the hosting units(s) to be tested.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

object[]

An array of PSObjects which pair the name and availability of the name

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

Test-HypHostingUnitNameAvailable -HostingUnitName \$NewHostingUnitName

This tests whether the value of \$NewHostingUnitName is unique or not, and can be used to create a new hosting unit or rename an existing one without failing. True is returned if the name is unique.



# Test-HypervisorConnectionNameAvailable

Jan 04, 2017

Checks to ensure that the proposed name for a hypervisor connection is unused.

## Syntax

```
Test-HypervisorConnectionNameAvailable -HypervisorConnectionName <String[]> [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Use this command to check that the proposed name for a hypervisor connection is unused. This check is done without regard for scoping of the existing hypervisor connection, so the names of inaccessible hypervisor connection are also checked.

## Related topics

New-Item

Rename-Item

## Parameters

**-HypervisorConnectionName**<String[]>

The name or names of the hypervisor connection(s) to be tested.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

object[]

An array of PSObjects which pair the name and availability of the name

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

Test-HypHypervisorConnectionNameAvailable -HypervisorConnectionName \$NewConnectionName

This tests whether the value of \$NewConnectionName is unique or not, and can be used to create a new hypervisor connection or rename an existing one. True is returned if the name is unique.

# Update-HypHypervisorConnection

Jan 04, 2017

Requests the host service to update the connection properties that depend on the version of hypervisor in use.

## Syntax

```
Update-HypHypervisorConnection [-LiteralPath] <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Use this command to update the version-specific properties of a hypervisor connection, after an upgrade to the hypervisor system which may provide new capabilities.

## Related topics

### Parameters

#### **-LiteralPath**<String>

Specifies the path within a Host Service provider to the hypervisor connection item to be updated. The path specified must be in one of the following formats; <drive>:\Connections\<HypervisorConnectionName> or <drive>:\Connections\{HypervisorConnection Uid}

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

#### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in will connect to. This can be provided as a

host name or an IP address.

Required?	false
Default Value	LocalHost. Once a value is provided by any cmdlet, this value will become the default.
Accept Pipeline Input?	false

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### InvalidPath

The path provided is not in the required format.

#### HypervisorInMaintenanceMode

The hypervisor is in maintenance mode and cannot be updated.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Update-HypervisorConnection -LiteralPath xdhyp:\connections\Connection1
```

This command requests that the properties of Connection1 be updated to match the current capabilities of the hypervisor.

# Citrix.MachineCreation.Admin.V2

Jan 04, 2017  
Overview

Name	Description
<a href="#">ProvMachineCreationSnapin</a>	The Machine Creation Service PowerShell snap-in provides administrative
<a href="#">Prov Filtering</a>	Describes the common filtering options for XenDesktop cmdlets.
<a href="#">providers</a>	Describes how Windows PowerShell providers provide access to data and

## Cmdlets

Name	Description
<a href="#">Add-ProvSchemeControllerAddress</a>	Adds a list of host names (as DNS addresses) to a provisioning scheme.
<a href="#">Add-ProvSchemeMetadata</a>	Adds metadata on the given ProvisioningScheme.
<a href="#">Add-ProvSchemeScope</a>	Add the specified ProvisioningScheme(s) to the given scope(s).
<a href="#">Add-ProvTaskMetadata</a>	Adds metadata on the given Task.
<a href="#">Get-ProvDBConnection</a>	Gets the database string for the specified data store used by the MachineCreation Service.
<a href="#">Get-ProvDBSchema</a>	Gets a script that creates the MachineCreation Service database schema for the specified data store.
<a href="#">Get-ProvDBVersionChangeScript</a>	Gets a script that updates the MachineCreation Service database schema.
<a href="#">Get-ProvInstalledDBVersion</a>	Gets a list of all available database schema versions for the MachineCreation Service.
<a href="#">Get-ProvObjectReference</a>	Returns the number of local objects holding references to objects from other services.
<a href="#">Get-ProvScheme</a>	Gets the list of provisioning schemes.
<a href="#">Get-ProvSchemeMasterVMImageHistory</a>	Gets the list of master VM snapshots that have been used to provide hard disks to provisioning schemes.

Name	Description
Get-ProvScopedObject	Gets the details of the scoped objects for the MachineCreation Service.
Get-ProvService	Gets the service record entries for the MachineCreation Service.
Get-ProvServiceAddedCapability	Gets any added capabilities for the MachineCreation Service on the controller.
Get-ProvServiceConfigurationData	Gets configuration data for the service.
Get-ProvServiceInstance	Gets the service instance entries for the MachineCreation Service.
Get-ProvServiceStatus	Gets the current status of the MachineCreation Service on the controller.
Get-ProvTask	Gets the task history for the MachineCreation Service.
Get-ProvVM	Gets the VMs that were created using Citrix XenDesktop Machine Creation Services.
Lock-ProvVM	Locks a VM.
New-ProvScheme	Creates a new provisioning scheme.
New-ProvVM	Creates a new virtual machine.
Publish-ProvMasterVmlImage	Update the master image associated with the provisioning scheme.
Remove-ProvScheme	Removes a provisioning scheme
Remove-ProvSchemeControllerAddress	Removes metadata from a provisioning scheme.
Remove-ProvSchemeMasterVmlImageHistory	Removes the history of provisioning scheme master image VMs.
Remove-ProvSchemeMetadata	Removes metadata from the given ProvisioningScheme.
Remove-ProvSchemeScope	Remove the specified ProvisioningScheme(s) from the given scope(s).
Remove-ProvServiceConfigurationData	Removes configuration data from the service.
Remove-ProvServiceMetadata	Removes metadata from the given Service.

Name	Description
Remove-ProvTask	Removes from the database completed tasks for the MachineCreation Service.
Remove-ProvTaskMetadata	Removes metadata from the given Task.
Remove-ProvVM	Removes virtual machines.
Rename-ProvScheme	Renames a provisioning scheme.
Reset- ProvServiceGroupMembership	Reloads the access permissions and configuration service locations for the MachineCreation Service.
Set-ProvDBConnection	Configures a database connection for the MachineCreation Service.
Set-ProvScheme	Changes the parameter values for a provisioning scheme.
Set-ProvSchemeMetadata	Adds or updates metadata on the given ProvisioningScheme.
Set-ProvServiceConfigurationData	Sets the value for the given key in the service configuration data.
Set-ProvServiceMetadata	Adds or updates metadata on the given Service.
Set-ProvTaskMetadata	Adds or updates metadata on the given Task.
Stop-ProvTask	Stops currently running MachineCreation Service tasks.
Switch-ProvTask	Moves all MachineCreation Service tasks from the current execution host to another.
Test-ProvDBConnection	Tests a database connection for the MachineCreation Service.
Test-ProvSchemeNameAvailable	Checks to ensure that the proposed name for a provisioning scheme is unused.
Unlock-ProvScheme	Unlocks a Provisioning Scheme.
Unlock-ProvVM	Unlocks a VM.



# about\_ProvMachineCreationSnapin

Jan 04, 2017

## TOPIC

about\_ProvMachineCreationSnapin

## SHORT DESCRIPTION

The Machine Creation Service PowerShell snap-in provides administrative functions for the Machine Creation Service.

## COMMAND PREFIX

All commands in this snap-in have 'Prov' in their name.

## LONG DESCRIPTION

The Machine Creation Service PowerShell snap-in enables both local and remote administration of the Machine Creation Service. It provides facilities to create virtual machines and manage the associated disk images.

The snap-in provides two main entities:

### Provisioning Scheme

Specifies details of new virtual machines created by the Machine Creation Service. Provisioning schemes define the following information.

#### Hosting Unit

Provides details of the hypervisor and storage on which new virtual machines will be created. Stored and maintained by the Host Service and PowerShell snap-in.

#### Identity Pool

Lists the Active Directory computer accounts available for use by new virtual machines. Stored and maintained by the Active Directory Identity Service and PowerShell snap-in.

#### Master Image

Specifies the disk image that will be used for new virtual machines. Accessed through the hosting provider in the Host Service snap-in.

### Provisioned VM

Defines the virtual machines created by the Machine Creation Services. These virtual machines are associated with the provisioning scheme from

which they were created.

The processes of creating provisioning schemes and new virtual machines can take a significant amount of time to complete. For this reason, these long-running tasks can be run asynchronously so that other commands are accessible while the processes are running. Note, however, that only one long-running task can operate on a provisioning scheme at any one time. The processes are monitored using the Get-ProvTask command. For more information, see the help for Get-ProvTask.

# about\_Prov\_Filtering

Jan 04, 2017

## TOPIC

XenDesktop - Advanced Dataset Filtering

## SHORT DESCRIPTION

Describes the common filtering options for XenDesktop cmdlets.

## LONG DESCRIPTION

Some cmdlets operate on large quantities of data and, to reduce the overhead of sending all of that data over the network, many of the Get- cmdlets support server-side filtering of the results.

The conventional way of filtering results in PowerShell is to pipeline them into Where-Object, Select-Object, and Sort-Object, for example:

```
Get-<Noun> | Where { $_.Size = 'Small' } | Sort 'Date' | Select -First 10
```

However, for most XenDesktop cmdlets the data is stored remotely and it would be slow and inefficient to retrieve large amounts of data over the network and then discard most of it. Instead, many of the Get- cmdlets provide filtering parameters that allow results to be processed on the server, returning only the required results.

You can filter results by most object properties using parameters derived from the property name. You can also sort results or limit them to a specified number of records:

```
Get-<Noun> -Size 'Small' -SortBy 'Date' -MaxRecordCount 10
```

You can express more complex filter conditions using a syntax and set of operators very similar to those used by PowerShell expressions.

Those cmdlets that support filtering have the following common parameters:

-MaxRecordCount <int>

Specifies the maximum number of results to return.  
For example, to return only the first nine results use:

```
Get-<Noun> -MaxRecordCount 9
```

If not specified, only the first 250 records are returned, and if more are available, a warning is produced:

WARNING: Only first 250 records returned. Use -MaxRecordCount to retrieve more.

You can suppress this warning by using -WarningAction or by specifying a value for -MaxRecordCount.

To retrieve all records, specify a large number for -MaxRecordCount. As the value is an integer, you can use the following:

```
Get-<Noun> -MaxRecordCount [int]::MaxValue
```

-ReturnTotalRecordCount [<SwitchParameter>]

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. For example:

```
Get-<Noun> -MaxRecordCount 9 -ReturnTotalRecordCount  
....
```

```
Get-<Noun> : Returned 9 of 10 items  
At line:1 char:18  
+ Get-<Noun> <<<< -MaxRecordCount 9 -ReturnTotalRecordCount  
+ CategoryInfo          : OperationStopped: (:) [Get-<Noun>], PartialDataException  
+ FullyQualifiedErrorId : PartialData,Citrix.<SDKName>.SDK.Get<Noun>
```

The count can be accessed using the TotalAvailableResultCount property:

```
$count = $error[0].TotalAvailableResultCount
```

-Skip <int>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

-SortBy <string>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order, respectively. Ascending order is assumed if no prefix is present.

Sorting occurs before -MaxRecordCount and -Skip parameters are applied. For example, to sort by Name and then by Count (largest first) use:

```
-SortBy 'Name,-Count'
```

By default, sorting by an enumeration property uses the numeric value of the elements. You can specify a different sort order by qualifying the name with an ordered list of elements or their numeric values, or <null> to indicate the placement of null values.

Elements not mentioned are placed at the end in their numeric order. For example, to sort by two different enums and then by the object id:

```
-SortBy 'MyState(StateC,<null>,StateA,StateB),Another(0,3,2,1),Id'
```

```
-Filter <String>
```

This parameter lets you specify advanced filter expressions, and supports combination of conditions with -and and -or, and grouping with braces. For example:

```
Get-<Noun> -Filter 'Name -like "High*" -or (Priority -eq 1 -and Severity -ge 2)'
```

The syntax is close enough to PowerShell syntax that you can use script blocks in most cases. This can be easier to read as it reduces quoting:

```
Get-<Noun> -Filter { Count -ne $null }
```

The full -Filter syntax is provided below.

## EXAMPLES

Filtering by strings performs a case-insensitive wildcard match.

Separate parameters are combined with an implicit -and operator.

Normal PowerShell quoting rules apply, so you can use single or double quotes, and omit the quotes altogether for many strings. The order of parameters does not make any difference. The following are equivalent:

```
Get-<Noun> -Company Citrix -Product Xen*
Get-<Noun> -Company "citrix" -Product '[X]EN*'
Get-<Noun> -Product "Xen*" -Company "CITRIX"
Get-<Noun> -Filter { Company -eq 'Citrix' -and Product -like 'Xen*' }
```

See [about\\_Quoting\\_Rules](#) and [about\\_Wildcards](#) for details about PowerShell handling of quotes and wildcards.

To avoid wildcard matching or include quote characters, you can escape the wildcards using the normal PowerShell escape mechanisms (see [about\\_Escape\\_Characters](#)), or switch to a filter expression and the `-eq` operator:

```
Get-<Noun> -Company "Abc[*]"           # Matches Abc*
Get-<Noun> -Company "Abc`*"           # Matches Abc*
Get-<Noun> -Filter { Company -eq "Abc*" } # Matches Abc*
Get-<Noun> -Filter { Company -eq "A`"B`C" } # Matches A"B'C
```

Simple filtering by numbers, booleans, and TimeSpans perform direct equality comparisons, although if the value is nullable you can also search for null values. Here are some examples:

```
Get-<Noun> -Uid 123
Get-<Noun> -Enabled $true
Get-<Noun> -Duration 1:30:40
Get-<Noun> -NullableProperty $null
```

More comparisons are possible using advanced filtering with `-Filter`:

```
Get-<Noun> -Filter 'Capacity -ge 10gb'
Get-<Noun> -Filter 'Age -ge 20 -and Age -lt 40'
Get-<Noun> -Filter 'VolumeLevel -like "[123]"'
Get-<Noun> -Filter 'Enabled -ne $false'
Get-<Noun> -Filter 'NullableProperty -ne $null'
```

You can check boolean values without an explicit comparison operator, and you can also combine them with `-not`:

```
Get-<Noun> -Filter 'Enabled' # Equivalent to 'Enabled -eq $true'
Get-<Noun> -Filter '-not Enabled' # Equivalent to 'Enabled -eq $false'
```

See [about\\_Comparison\\_Operators](#) for an explanation of the operators, but note that only a subset of PowerShell operators are supported (`-eq`, `-ne`, `-gt`, `-ge`, `-lt`, `-le`, `-like`, `-notlike`, `-in`, `-notin`, `-contains`, `-notcontains`).

Enumeration values can either be specified using typed values or the string name of the enumeration value:

```
Get-<Noun> -Shape [Shapes]::Square
Get-<Noun> -Shape Circle
```

With filter expressions, typed values can be specified with simple variables or quoted strings. They also support enumerations with wildcards:

```
$s = [Shapes]::Square
Get-<Noun> -Filter { Shape -eq $s -or Shape -eq "Circle" }
Get-<Noun> -Filter { Shape -like 'C*' }
```

By their nature, floating point values, DateTime values, and TimeSpan values are best suited to relative comparisons rather than just equality. DateTime strings are converted using the locale and time zone of the user device, but you can use ISO8601 format strings (YYYY-MM-DDThh:mm:ss.sTZD) to avoid ambiguity. You can also use standard PowerShell syntax to create these values:

```
Get-<Noun> -Filter { StartTime -ge "2010-08-23T12:30:00.0Z" }
$d = [DateTime]"2010-08-23T12:30:00.0Z"
Get-<Noun> -Filter { StartTime -ge $d }
$d = (Get-Date).AddDays(-1)
Get-<Noun> -Filter { StartTime -ge $d }
```

Relative times are quite common and, when using filter expressions, you can also specify DateTime values using a relative format:

```
Get-<Noun> -Filter { StartTime -ge '-2' } # Two days ago
Get-<Noun> -Filter { StartTime -ge '-1:30' } # Hour and a half ago
Get-<Noun> -Filter { StartTime -ge '-0:0:30' } # 30 seconds ago
```

## ARRAY PROPERTIES

When filtering against list or array properties, simple parameters perform a case-insensitive wildcard match against each of the members. With filter expressions, you can use the -contains and -notcontains operators. Unlike PowerShell, these perform wildcard matching on strings.

Note that for array properties the naming convention is for the returned property to be plural, but the parameter used to search for any match is singular. The following are equivalent (assuming Users is an array property):

```
Get-<Noun> -User Fred*
Get-<Noun> -Filter { User -like "Fred*" }
Get-<Noun> -Filter { Users -contains "Fred*" }
```

You can also use the singular form with -Filter to search using other operators:

```
# Match if any user in the list is called "Frederick"
Get-<Noun> -Filter { User -eq "Frederick" }
# Match if any user in the list has a name alphabetically below 'F'
Get-<Noun> -Filter { User -lt 'F' }
```

## COMPLEX EXPRESSIONS

When matching against multiple values, you can use a sequence of comparisons joined with -or operators, or you can use -in and -notin:

```
Get-<Noun> -Filter { Shape -eq 'Circle' -or Shape -eq 'Square' }
$shapes = 'Circle','Square'
Get-<Noun> -Filter { Shape -in $shapes }
$sides = 1..4
Get-<Noun> -Filter { Sides -notin $sides }
```

Braces can be used to group complex expressions, and override the default left-to-right evaluation of -and and -or. You can also use -not to invert the sense of any sub-expression:

```
Get-<Noun> -Filter { Size -gt 4 -or (Color -eq 'Blue' -and Shape -eq 'Circle') }
Get-<Noun> -Filter { Sides -lt 5 -and -not (Color -eq 'Blue' -and Shape -eq 'Circle') }
```

## PAGING

The simplest way to page through data is to use the -Skip and -MaxRecordCount parameters. So, to read the first three pages of data with 10 records per page, use:

```
Get-<Noun> -Skip 0 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 10 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 20 -MaxRecordCount 10 <other filtering criteria>
```

You must include the same filtering criteria on each call, and ensure that the data is sorted consistently.

The above approach is often acceptable, but as each call performs an independent query, data changes can result in records being skipped or appearing twice. One approach to improve this is to sort by a unique id field and then start the search for the next page at the unique id after the last unique id of the previous page. For example:

```
# Get the first page
Get-<Noun> -MaxRecordCount 10 -SortBy SerialNumber

SerialNumber ...
----- ---
A120004
A120007
... 7 other records ...
A120900

# Get the next page
Get-<Noun> -MaxRecordCount 10 -Filter { FirstName -gt 'A120900' }

SerialNumber ...
```



----- ---  
A120901  
B220000  
...

## FILTER SYNTAX DEFINITION

<Filter> ::= <ScriptBlock> | <ComponentList>

<ScriptBlock> ::= "{" <ComponentList> "}"

<ComponentList> ::= <Component> <AndOrOperator> <ComponentList> |

<Component>

<Component> ::= <NotOperator> <Factor> |

<Factor>

<Factor> ::= "(" <ComponentList> ")" |

<PropertyName> <ComparisonOperator> <Value> |  
<PropertyName>

<AndOrOperator> ::= "-and" | "-or"

<NotOperator> ::= "-not" | "!"

<ComparisonOperator>

::= "-eq" | "-ne" | "-le" | "-ge" | "-lt" | "-gt" |  
"-like" | "-notlike" | "-contains" | "-notcontains" |  
"-in" | "-notin"

<PropertyName> ::= <simple name of property>

<Value> ::= <string literal> | <numeric literal> |

<scalar variable> | <array variable> |  
"\$null" | "\$true" | "\$false"

Numeric literals support decimal and hexadecimal literals, with optional multiplier suffixes (kb, mb, gb, tb, pb).

Dates and times can be specified as string literals. The current culture determines what formats are accepted. To avoid any ambiguity, use strings formatted to the ISO8601 standard. If not specified, the current time zone is used.

Relative date-time string literals are also supported, using a minus sign followed by a TimeSpan. For example, "-1:30" means 1 hour and 30 minutes ago.

# about\_providers

Jan 04, 2017

## TOPIC

about\_Providers

## SHORT DESCRIPTION

Describes how Windows PowerShell providers provide access to data and components that would not otherwise be easily accessible at the command line. The data is presented in a consistent format that resembles a file system drive.

## LONG DESCRIPTION

Windows PowerShell providers are Microsoft .NET Framework-based programs that make the data in a specialized data store available in Windows PowerShell so that you can view and manage it.

The data that a provider exposes appears in a drive, and you access the data in a path like you would on a hard disk drive. You can use any of the built-in cmdlets that the provider supports to manage the data in the provider drive. And, you can use custom cmdlets that are designed especially for the data.

The providers can also add dynamic parameters to the built-in cmdlets. These are parameters that are available only when you use the cmdlet with the provider data.

## BUILT-IN PROVIDERS

Windows PowerShell includes a set of built-in providers that you can use to access the different types of data stores.

Provider Drive Data store ----- Alias Alias: Windows PowerShell aliases

Certificate Cert: x509 certificates for digital signatures

Environment Env: Windows environment variables

FileSystem \* File system drives, directories, and files

Function Function: Windows PowerShell functions

Registry HKLM:, HKCU Windows registry

Variable Variable: Windows PowerShell variables

WS-Management WSMAN WS-Management configuration information

\* The FileSystem drives vary on each system.

You can also create your own Windows PowerShell providers, and you can install providers that others develop. To list the providers that are available in your session, type:

```
get-psprovider
```

## INSTALLING AND REMOVING PROVIDERS

Windows PowerShell providers are delivered to you in Windows PowerShell snap-ins, which are .NET Framework-based programs that are compiled into .dll files. The snap-ins can include providers and cmdlets.

Before you use the provider features, you have to install the snap-in and then add it to your Windows PowerShell session. For more information, see `about_PsSnapins`.

You cannot uninstall a provider, although you can remove the Windows PowerShell snap-in for the provider from the current session. If you do, you will remove all the contents of the snap-in, including its cmdlets.

To remove a provider from the current session, use the `Remove-PsSnapin` cmdlet. This cmdlet does not uninstall the provider, but it makes the provider unavailable in the session.

You can also use the `Remove-PsDrive` cmdlet to remove any drive from the current session. This data on the drive is not affected, but the drive is no longer available in that session.

## VIEWING PROVIDERS

To view the Windows PowerShell providers on your computer, type:

```
get-psprovider
```

The output lists the built-in providers and the providers that you added to the session.

## THE PROVIDER CMDLETS

The following cmdlets are designed to work with the data exposed by any provider. You can use the same cmdlets in the same way to manage the different types of data that providers expose. After you learn to manage the data of one provider, you can use the same procedures with the data from any provider.

For example, the `New-Item` cmdlet creates a new item. In the C: drive that is supported by the `FileSystem` provider, you can use `New-Item` to create a new file or folder. In the drives that are supported by the `Registry` provider, you can use `New-Item` to create a new registry key. In the `Alias:` drive, you can use `New-Item` to create a new alias.

For detailed information about any of the following cmdlets, type:

```
get-help <cmdlet-name> -detailed
```

## CHILDITEM CMDLETS

```
Get-ChildItem
```

## CONTENT CMDLETS

```
Add-Content
```

Clear-Content  
Get-Content  
Set-Content

#### ITEM CMDLETS

Clear-Item  
Copy-Item  
Get-Item  
Invoke-Item  
Move-Item  
New-Item  
Remove-Item  
Rename-Item  
Set-Item

#### ITEMPROPERTY CMDLETS

Clear-ItemProperty  
Copy-ItemProperty  
Get-ItemProperty  
Move-ItemProperty  
New-ItemProperty  
Remove-ItemProperty  
Rename-ItemProperty  
Set-ItemProperty

#### LOCATION CMDLETS

Get-Location  
Pop-Location  
Push-Location  
Set-Location

#### PATH CMDLETS

Join-Path  
Convert-Path  
Split-Path  
Resolve-Path  
Test-Path

## PSDRIVE CMDLETS

- Get-PSDrive
- New-PSDrive
- Remove-PSDrive

## PSPROVIDER CMDLETS

- Get-PSPProvider

## VIEWING PROVIDER DATA

The primary benefit of a provider is that it exposes its data in a familiar and consistent way. The model for data presentation is a file system drive.

To use data that the provider exposes, you view it, move through it, and change it as though it were data on a hard drive. Therefore, the most important information about a provider is the name of the drive that it supports.

The drive is listed in the default display of the Get-PsProvider cmdlet, but you can get information about the provider drive by using the Get-PsDrive cmdlet. For example, to get all the properties of the Function: drive, type:

```
get-psdrive Function | format-list *
```

You can view and move through the data in a provider drive just as you would on a file system drive.

To view the contents of a provider drive, use the Get-Item or Get-ChildItem cmdlets. Type the drive name followed by a colon (.). For example, to view the contents of the Alias: drive, type:

```
get-item alias:
```

You can view and manage the data in any drive from another drive by including the drive name in the path. For example, to view the HKLM\Software registry key in the HKLM: drive from another drive, type:

```
get-childitem hklm:\software
```

To open the drive, use the Set-Location cmdlet. Remember the colon when you specify the drive path. For example, to change your location to the root directory of the Cert: drive, type:

```
set-location cert:
```

Then, to view the contents of the Cert: drive, type:

```
get-childitem
```

## MOVING THROUGH HIERARCHICAL DATA

You can move through a provider drive just as you would a hard disk drive. If the data is arranged in a hierarchy of items within items, use a backslash (\) to indicate a child item. Use the following format:

```
drive:\location\child-location\...
```

For example, to change your location to the HKLM\Software registry key, type a Set-Location command, such as:

```
set-location hklm:\software
```

You can also use relative references to locations. A dot (.) represents the current location. For example, if you are in the HKLM:\Software\Microsoft registry key, and you want to list the registry subkeys in the HKLM:\Software\Microsoft\PowerShell key, type the following command:

```
get-childitem .powershell
```

## FINDING DYNAMIC PARAMETERS

Dynamic parameters are cmdlet parameters that are added to a cmdlet by a provider. These parameters are available only when the cmdlet is used with the provider that added them.

For example, the Cert: drive adds the CodeSigningCert parameter to the Get-Item and Get-ChildItem cmdlets. You can use this parameter only when you use Get-Item or Get-ChildItem in the Cert: drive.

For a list of the dynamic parameters that a provider supports, see the Help file for the provider. Type:

```
get-help <provider-name>
```

For example:

```
get-help certificate
```

## LEARNING ABOUT PROVIDERS

Although all provider data appears in drives, and you use the same methods to move through them, the similarity stops there. The data stores that the provider exposes can be as varied as Active Directory locations and Microsoft Exchange Server mailboxes.

For information about individual Windows PowerShell providers, type:

```
get-help <ProviderName>
```

For example:

```
get-help registry
```

For a list of Help topics about the providers, type:

```
get-help * -category provider
```

SEE ALSO

about\_Locations

about\_Path\_Syntax



# Add-ProvSchemeControllerAddress

Jan 04, 2017

Adds a list of host names (as DNS addresses) to a provisioning scheme.

## Syntax

```
Add-ProvSchemeControllerAddress [-ProvisioningSchemeName] <String> [-ControllerAddress] <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ProvSchemeControllerAddress -ProvisioningSchemeUid <Guid> [-ControllerAddress] <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to associate controller hosts (and hence implicitly a set of brokers) with a specific provisioning scheme. This optional data is passed to the machines created by the Machine Creation Services, where it is used to associate the newly created machine with a broker. The list is returned along with the provisioning scheme that it is assigned to.

## Related topics

[Get-ProvScheme](#)

[Remove-ProvSchemeControllerAddress](#)

## Parameters

**-ProvisioningSchemeName**<String>

The name for the provisioning scheme that the list of addresses is to be added to.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ProvisioningSchemeUid**<Guid>

The unique identifier for the provisioning scheme that the list of addresses is to be added to.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-ControllerAddress**<String[]>

Specifies the array of DNS names to be added to the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

#### Input Type

Citrix.MachineCreation.Sdk.ProvisioningScheme You can pipe an object containing a parameter called 'ProvisioningSchemeName' to Add-ProvSchemeControllerAddress.

#### Return Values

Citrix.MachineCreation.Sdk.ProvisioningScheme

Add-ProvSchemeControllerAddress returns the updated ProvisioningScheme object containing the union of the old and new controller address lists.

ProvisioningSchemeUid <Guid>

The unique identifier for the provisioning scheme.

ProvisioningSchemeName <string>

The name of the provisioning scheme.

CpuCount <int>

The number of processors that VMs will be created with when using this scheme.

MemoryMB <int>

The maximum amount of memory that VMs will be created with when using this scheme.

MasterImageVM <string>

The path within the hosting unit provider to the VM or snapshot of which the scheme is currently using a copy.

MasterImageVMDate <DateTime>

The date and time that the copy of the VM image was made for the scheme.

IdentityPoolUid <Guid>

The unique identifier of the identity pool (from the ADIdentity PowerShell snap-in) that the scheme uses.

IdentityPoolName <string>

The name of the identity pool (from the ADIdentity PowerShell snap-in) that the scheme uses.

HostingUnitUid <Guid>

The unique identifier of the hosting unit (from the Hosting Unit PowerShell snap-in) that the scheme will use.

HostingUnitName <string>

The name of the hosting unit (from the Hosting Unit PowerShell snap-in) that the scheme will use.

CleanOnBoot <Boolean>

Indicates whether or not the VMs created are to be reset to a clean state on each boot.

TaskId <Guid>

The identifier of any current task that is running for the provisioning scheme.

Metadata <Citrix.MachineCreation.Sdk.Metadata[]>

The metadata associated with this provisioning scheme.

ControllerAddress <string[]>

The DNS names of the controllers associated with this provisioning scheme for Quick Deploy purposes.

#### Notes

In the case of failure, the following errors can result.

#### Error Codes

-----

#### ProvisioningSchemeNotFound

The specified provisioning scheme could not be located.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
C:\PS>Add-ProvSchemeControllerAddress -ProvisioningSchemeUid "01a4a008-8ce8-4165-ba9c-cdf15a6b0501" -ControllerAddress (ddcA.citrix.com,ddcB.citrix.com,ddcC.citrix2.com)
```

```
ProvisioningSchemeUid : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501
ProvisioningSchemeName : Scheme2
CpuCount              : 1
MemoryMB              : 1024
MasterImageVM         : Base.vm/Base.snapshot
MasterImageVMDate     : 17/05/2010 09:53:40
IdentityPoolUid       : 03743136-e43b-4a87-af74-ab71686b3c16
IdentityPoolName      : idPool1
HostingUnitUid        : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501
HostingUnitName       : HostUnit1
CleanOnBoot           : True
TaskId                : 00000000-0000-0000-0000-000000000000
Metadata              : {}
ControllerAddress     : {ddcA.citrix.com,ddcB.citrix.com,ddcC.citrix2.com}
Add a set of controllers to the provisioning scheme with the identifier "01a4a008-8ce8-4165-ba9c-cdf15a6b0501".
```

##### ----- EXAMPLE 2 -----

```
C:\PS>Get-ProvScheme -ProvisioningSchemeName scheme1 | Add-ProvSchemeControllerAddress -ControllerAddress (ddcA.citrix.com,ddcB.citrix.com,ddcC.citrix2.com)
```

```
ProvisioningSchemeUid : 7585f0de-192e-4847-a6d8-22713c3a2f42
ProvisioningSchemeName : Scheme1
CpuCount              : 1
MemoryMB              : 1024
MasterImageVM         : Base.vm/Base.snapshot
MasterImageVMDate     : 17/05/2010 09:53:40
IdentityPoolUid       : 03743136-e43b-4a87-af74-ab71686b3c16
IdentityPoolName      : idPool1
HostingUnitUid        : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501
HostingUnitName       : HostUnit1
CleanOnBoot           : True
TaskId                : 00000000-0000-0000-0000-000000000000
```

Metadata : {}

ControllerAddress : {ddcA.citrix.com,ddcB.citrix.com,ddcC.citrix2.com}

Add controller addresses to a provisioning scheme using a ProvisioningScheme object.

# Add-ProvSchemeMetadata

Jan 04, 2017

Adds metadata on the given ProvisioningScheme.

## Syntax

```
Add-ProvSchemeMetadata [-ProvisioningSchemeUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ProvSchemeMetadata [-ProvisioningSchemeUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ProvSchemeMetadata [-ProvisioningSchemeName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ProvSchemeMetadata [-ProvisioningSchemeName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ProvSchemeMetadata [-InputObject] <ProvisioningScheme[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ProvSchemeMetadata [-InputObject] <ProvisioningScheme[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given ProvisioningScheme objects. This cmdlet will not overwrite existing metadata on an object - use the Set-ProvSchemeMetadata cmdlet instead.

## Related topics

[Set-ProvSchemeMetadata](#)

[Remove-ProvSchemeMetadata](#)

## Parameters

**-ProvisioningSchemeUid**<Guid>

Id of the ProvisioningScheme

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-ProvisioningSchemeName**<String>

Name of the ProvisioningScheme

Required?	true
-----------	------

Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<ProvisioningScheme[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the ProvisioningScheme specified. The property cannot contain any of the following characters \/:#.\*?=<>|[]()"

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Value**<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

Citrix.MachineCreation.Sdk.Metadata

Add-ProvSchemeMetadata returns an array of objects containing the new definition of the metadata.

\n Property <string>

\n Specifies the name of the property.

\n Value <string>

\n Specifies the value for the property.

### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DuplicateObject

One of the specified metadata already exists.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

### Examples

#### ----- EXAMPLE 1 -----

```
c:\PS>Add-ProvSchemeMetadata -ProvisioningSchemeUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Property	Value
property	value

Add metadata with a name of 'property' and a value of 'value' to the ProvisioningScheme with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.



# Add-ProvSchemeScope

Jan 04, 2017

Add the specified ProvisioningScheme(s) to the given scope(s).

## Syntax

```
Add-ProvSchemeScope [-Scope] <String[]> -InputObject <ProvisioningScheme[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ProvSchemeScope [-Scope] <String[]> -ProvisioningSchemeUid <Guid[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-ProvSchemeScope [-Scope] <String[]> -ProvisioningSchemeName <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The AddProvSchemeScope cmdlet is used to associate one or more ProvisioningScheme objects with given scope(s).

There are multiple parameter sets for this cmdlet, allowing you to identify the ProvisioningScheme objects in different ways:

- ProvisioningScheme objects can be piped in or specified by the InputObject parameter
- The ProvisioningSchemeUid parameter specifies objects by ProvisioningSchemeUid
- The ProvisioningSchemeName parameter specifies objects by ProvisioningSchemeName (supports wildcards)

To add a ProvisioningScheme to a scope you need permission to change the scopes of the ProvisioningScheme and permission to add objects to all of the scopes you have specified.

If the ProvisioningScheme is already in a scope, that scope will be silently ignored.

## Related topics

[Remove-ProvSchemeScope](#)

[Get-ProvScopedObject](#)

## Parameters

**-Scope**<String[]>

Specifies the scopes to add the objects to.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-InputObject**<ProvisioningScheme[]>

Specifies the ProvisioningScheme objects to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-ProvisioningSchemeUid**<Guid[]>

Specifies the ProvisioningScheme objects to be added by ProvisioningSchemeUid.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-ProvisioningSchemeName**<String[]>

Specifies the ProvisioningScheme objects to be added by ProvisioningSchemeName.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

None

### Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### ScopeNotFound

One of the specified scopes was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command with the specified objects or scopes.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Add-ProvSchemeScope Finance -ProvisioningSchemeUid 6702C5D0-C073-4080-A0EE-EC74CB537C52
```

Adds a single ProvisioningScheme to the 'Finance' scope.

### ----- EXAMPLE 2 -----

```
c:\PS>Add-ProvSchemeScope Finance,Marketing -ProvisioningSchemeUid 6702C5D0-C073-4080-A0EE-EC74CB537C52
```

Adds a single ProvisioningScheme to the multiple scopes.

### ----- EXAMPLE 3 -----

```
c:\PS>Get-ProvScheme | Add-ProvSchemeScope Finance
```

Adds all visible ProvisioningScheme objects to the 'Finance' scope.

### ----- EXAMPLE 4 -----

```
c:\PS>Add-ProvSchemeScope Finance -ProvisioningSchemeName A*
```

Adds ProvisioningScheme objects with a name starting with an 'A' to the 'Finance' scope.

# Add-ProvTaskMetadata

Jan 04, 2017

Adds metadata on the given Task.

## Syntax

```
Add-ProvTaskMetadata [-TaskId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Add-ProvTaskMetadata [-TaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Add-ProvTaskMetadata [-InputObject] <Task[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Add-ProvTaskMetadata [-InputObject] <Task[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

## Detailed Description

Use this cmdlet to store additional custom data against given Task objects. This cmdlet does not overwrite existing metadata on an object - use the Set-ProvTaskMetadata cmdlet instead.

## Related topics

[Set-ProvTaskMetadata](#)

[Remove-ProvTaskMetadata](#)

[Get-ProvTask](#)

[Stop-ProvTask](#)

[Remove-ProvTask](#)

[Switch-ProvTask](#)

## Parameters

**-TaskId**<Guid>

Id of the Task

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<Task[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Task specified. The property cannot contain any of the following characters \/:#.\*?=<>|[]()"

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Value**<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

Citrix.MachineCreation.Sdk.Metadata

Add-ProvTaskMetadata returns an array of objects containing the new definition of the metadata.

Property <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

### Notes

If the command fails, the following errors can result.

Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DuplicateObject

One of the specified metadata already exists.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Add-ProvTaskMetadata -TaskId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Property	Value
property	value

Add metadata with a name of 'property' and a value of 'value' to the Task with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.



# Get-ProvDBConnection

Jan 04, 2017

Gets the database string for the specified data store used by the MachineCreation Service.

## Syntax

```
Get-ProvDBConnection [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the database connection string for the specified data store.

If the returned string is blank, no valid connection string has been specified. In this case the service is running, but is idle and awaiting specification of a valid connection string.

## Related topics

[Get-ProvServiceStatus](#)

[Set-ProvDBConnection](#)

[Test-ProvDBConnection](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

system.string

The database connection string configured for the MachineCreation Service.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

NoDBConnections

The database connection string for the MachineCreation Service has not been specified.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ProvDBConnection
```

Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True  
Get the database connection string for the MachineCreation Service.

# Get-ProvDBSchema

Jan 04, 2017

Gets a script that creates the MachineCreation Service database schema for the specified data store.

## Syntax

```
Get-ProvDBSchema [-DatabaseName <String>] [-ServiceGroupName <String>] [-ScriptType <ScriptTypes>] [-LocalDatabase] [-Sid <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets SQL scripts that can be used to create a new MachineCreation Service database schema, add a new MachineCreation Service to an existing site, remove a MachineCreation Service from a site, or create a database server logon for a MachineCreation Service. If no Sid parameter is provided, the scripts obtained relate to the currently selected MachineCreation Service instance, otherwise the scripts relate to MachineCreation Service instance running on the machine identified by the Sid provided. When obtaining the Evict script, a Sid parameter must be supplied. The current service instance is that on the local machine, or that explicitly specified by the last usage of the -AdminAddress parameter to a MachineCreation SDK cmdlet. The service instance used to obtain the scripts does not need to be a member of a site or to have had its database connection configured. The database scripts support only Microsoft SQL Server, or SQL Server Express, and require Windows integrated authentication to be used. They can be run using SQL Server's SQLCMD utility, or by copying the script into an SQL Server Management Studio (SSMS) query window and executing the query. If using SSMS, the query must be executed in 'SMDCMD mode'. The ScriptType parameter determines which script is obtained. If ScriptType is not specified, or is FullDatabase, the script contains:

- o Creation of service schema
- o Creation of database server logon
- o Creation of database user
- o Addition of database user to MachineCreation Service roles

If ScriptType is Instance, the returned script contains:

- o Creation of database server logon
- o Creation of database user
- o Addition of database user to MachineCreation Service roles

If ScriptType is Evict, the returned script contains:

- o Removal of MachineCreation Service instance from database
- o Removal of database user

If ScriptType is Login, the returned script contains:

- o Creation of database server logon only

If the service uses two data stores they can exist in the same database. You do not need to configure a database before

using this command.

## Related topics

[Set-ProvDBConnection](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database for which the schema will be generated.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ServiceGroupName**<String>

Specifies the name of the service group to be used when creating the database schema. The service group is a collection of all the MachineCreation services that share the same database instance and are considered equivalent; that is, all the services within a service group can be used interchangeably.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Script Type**<ScriptTypes>

Specifies the type of database script returned. Available script types are:

### Database

Returns a full database script that can be used to create a database schema for the MachineCreation Service in a database instance that does not already contain a schema for this service. The DatabaseName and ServiceGroupName parameters must be specified to create a script of this type.

### Instance

Returns a permissions script that can be used to add further MachineCreation services to an existing database instance that already contains the full MachineCreation service schema, associating the services to the Service Group. The Sid parameter can optionally be specified to create a script of this type.

### Login

Returns a database logon script that can be used to add the required logon accounts to an existing database instance that contains the MachineCreation Service schema. This is used primarily when creating a mirrored database environment. The DatabaseName parameter must be specified to create a script of this type.

**Evict**

Returns a script that can be used to remove the specified MachineCreation Service from the database entirely. The DatabaseName and Sid parameters must be specified to create a script of this type.

Required?	false
Default Value	Database
Accept Pipeline Input?	false

**-LocalDatabase**<SwitchParameter>

Specifies whether the database script is to be used in a database instance run on the same controller as other services in the service group. Including this parameter ensures the script creates only the required permissions for local services to access the database schema for MachineCreation services.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Sid**<String>

Specifies the SID of the controller on which the MachineCreation Service instance to remove from the database is running.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByValue)

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
-----------	-------

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

### Systemstring

A string containing the required SQL script for application to a database.

## Notes

The scripts returned support Microsoft SQL Server Express Edition, Microsoft SQL Server Standard Edition, and Microsoft SQL Server Enterprise Edition databases only, and are generated on the assumption that integrated authentication will be used.

If the ScriptType parameter is not included or set to 'FullDatabase', the full database script is returned, which will:

Create the database schema.

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist).

If the ScriptType parameter is set to 'Instance', the script will:

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist) and associate it with a user.

If the ScriptType parameter is set to 'Login', the script will:

Create the logon (providing the schema does not already exist) and associate it with a pre-existing user of the same name.

If the LocalDatabase parameter is included, the NetworkService account will be added to the list of accounts permitted to access the database. This is required only if the database is run on a controller.

If the command fails, the following errors can be returned.

### Error Codes

-----

#### GetSchemasFailed

The database schema could not be found.

#### ActiveDirectoryAccountResolutionFailed

The specified Active Directory account or Group could not be found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ProvDBSchema -DatabaseName MyDB -ServiceGroupName MyServiceGroup > c:\ProvSchema.sql
Get the full database schema for site data store of the MachineCreation Service and copy it to a file called
'c:\ProvSchema.sql'.
```

This script can then be used to create the schema in a pre-existing database named 'MyDB' that does not already contain a MachineCreation Service site schema.

----- **EXAMPLE 2** -----

```
c:\PS>Get-ProvDBSchema -DatabaseName MyDB -scriptType Login > c:\MachineCreationLogins.sql
Get the logon scripts for the MachineCreation Service.
```

# Get-ProvDBVersionChangeScript

Jan 04, 2017

Gets a script that updates the MachineCreation Service database schema.

## Syntax

```
Get-ProvDBVersionChangeScript -DatabaseName <String> -TargetVersion <Version> [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Returns a database script that can be used to upgrade or downgrade the site or secondary schema for the MachineCreation Service from the current schema version to a different version.

## Related topics

[Get-ProvInstalledDBVersion](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database instance to which the update applies.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-TargetVersion**<Version>

Specifies the version of the database you want to update to.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.



Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

System.Management.Automation.PSObject

A PSObject containing the required SQL script for application to a database.

## Notes

The PSObject returned by this cmdlet contains the following properties:

- Script The raw text of the SQL script to apply the update, or null in the case when no upgrade path to the specified target version exists.
- NeedExclusiveAccess Indicates whether all services in the service group must be shut down during the update or not.
- CanUndo Indicates whether the generated script allows the updated schema to be reverted to the state prior to the update.

Scripts to update the schema version are stored in the database so any service in the service group can obtain these scripts. Extreme caution should be exercised when using update scripts. Citrix recommends backing up the database before attempting to upgrade the schema. Database update scripts may require exclusive use of the schema and so may not be able to execute while any MachineCreation services are running. However, this depends on the specific update being carried out.

After a schema update has been carried out, services that require the previous version of the schema may cease to operate. The ServiceState parameter reported by the Get-ProvServiceStatus command provides information about service compatibility. For example, if the schema has been upgraded to a more recent version that a service cannot use, the service reports "DBNewerVersionThanService".

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoOp

The operation was successful but had no effect.

#### NoDBConnections

The database connection string for the MachineCreation Service has not been specified.

#### DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $update = Get-ProvDBVersionChangeScript -DatabaseName MyDb -TargetVersion 1.0.75.0
```

```
C:\PS> $update.Script > update_75.sql
```

Gets an SQL update script to update the current schema to version 1.0.75.0. The resulting update\_75.sql script is suitable for direct use with the SQL Server SQLCMD utility.

# Get-ProvInstalledDBVersion

Jan 04, 2017

Gets a list of all available database schema versions for the MachineCreation Service.

## Syntax

```
Get-ProvInstalledDBVersion [-Upgrade] [-Downgrade] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the current version of the MachineCreation Service database schema, if no flags are set, otherwise returns versions for which upgrade or downgrade scripts are available and have been stored in the database.

## Related topics

## Parameters

**-Upgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be updated should be returned.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Downgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be reverted should be returned.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.

Accept Pipeline Input?	false
------------------------	-------

## Return Values

### System.Version

The Get-ProvInstalledDbVersion command returns objects containing the new definition of the MachineCreation Service database schema version.

Major <Integer>

Minor <Integer>

Build <Integer>

Revision <Integer>

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

Both the Upgrade and Downgrade flags were specified.

#### NoOp

The operation was successful but had no effect.

#### NoDBConnections

The database connection string for the MachineCreation Service has not been specified.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-ProvInstalledDBVersion
```

```
Major Minor Build Revision
```

```
-----
```

```
5 6 0 0
```

Get the currently installed version of the MachineCreation Service database schema.

### ----- EXAMPLE 2 -----

```
c:\PS>Get-ProvInstalledDBVersion -Upgrade
```

```
Major Minor Build Revision
```

```
-----
```

```
6 0 0 0
```

Get the versions of the MachineCreation Service database schema for which upgrade scripts are supplied.

# Get-ProvObjectReference

Jan 04, 2017

Returns the number of local objects holding references to objects from other services.

## Syntax

```
Get-ProvObjectReference [-HostingUnitUid <Guid[]>] [-IdentityPoolUid <Guid[]>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Returns for each hosting unit or identity pool GUID the number of references by provisioning schemes or by long running task. This check is done without regard for scoping of existing provisioning schemes, references by inaccessible schemes are also checked.

## Related topics

## Parameters

**-HostingUnitUid**<Guid[]>

The identifiers of the hosting units(s) to be tested.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-IdentityPoolUid**<Guid[]>

The identifiers of the identity pool(s) to be tested.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
-----------	-------

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

### ObjectReferenceCount

An object which contain the input object identifier, its type, the type of referencing object and the number of references.

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
Get-ProvObjectReference -HostingUnitUid 5B66A060-85E1-4DBD-9D1B-BF79881D3BB1
```

```
Count      : 1
```

ObjectId : 5b66a060-85e1-4dbd-9d1b-bf79881d3bb1  
Source : ProvisioningScheme  
Target : HostingUnit

Count : 0  
ObjectId : 5b66a060-85e1-4dbd-9d1b-bf79881d3bb1  
Source : Task  
Target : HostingUnit

This checks a single hosting unit for objects that have references to them; in this case we only have a single provisioning scheme that relates to it.



# Get-ProvScheme

Jan 04, 2017

Gets the list of provisioning schemes.

## Syntax

```
Get-ProvScheme [[-ProvisioningSchemeName] <String>] [-ProvisioningSchemeUid <Guid>] [-Scopeld <Guid>] [-ScopeName <String>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Lets you retrieve the list of defined provisioning schemes.

## Related topics

[New-ProvScheme](#)

[Remove-ProvScheme](#)

[Add-ProvSchemeMetadata](#)

[Remove-ProvSchemeMetadata](#)

[Add-ProvSchemeControllerAddress](#)

[Remove-ProvSchemeControllerAddress](#)

## Parameters

**-ProvisioningSchemeName**<String>

The name of the provisioning scheme.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ProvisioningSchemeUid**<Guid>

The unique identifier of the provisioning scheme.

Required?	false
Default Value	

Accept Pipeline Input?	false
------------------------	-------

**-ScopeId**<Guid>

Gets only results with a scope matching the specified scope identifier.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ScopeName**<String>

Gets only results with a scope matching the specified scope name.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ReturnTotalRecordCount**<SwitchParameter>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-MaxRecordCount**<Int32>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-Skip**<Int32>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-SortBy**<String>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Filter**<String>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	LocalHost. When a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.MachineCreation.Sdk.ProvisioningScheme

This object provides details of the provisioning scheme and contains the following information:

ProvisioningSchemeUid <Guid>

The unique identifier for the provisioning scheme.

ProvisioningSchemeName <string>

The name of the provisioning scheme.

CpuCount <int>

The number of processors that VMs will be created with when using this scheme.

MemoryMB <int>

The maximum amount of memory that VMs will be created with when using this scheme.

MasterImageVM <string>

The path within the hosting unit provider to the copy of the VM snapshot that the scheme uses.

MasterImageVMDate <DateTime>

The date and time that the copy was made of the VM snapshot used by the scheme.

IdentityPoolUid <Guid>

The unique identifier of the identity pool (from the ADIdentity PowerShell snap-in) that the scheme uses.

IdentityPoolName <string>

The name of the identity pool (from the ADIdentity PowerShell snap-in) that the scheme uses.

HostingUnitUid <Guid>

The unique identifier of the hosting unit (from the Hosting Unit PowerShell snap-in) that the scheme uses.

HostingUnitName <string>

The name of the hosting unit (from the Hosting Unit PowerShell snap-in) that the scheme uses.

CleanOnBoot <Boolean>

Indicates whether the VMs that are created will be reset to a clean state on each boot.

TaskId <Guid>

The identifier of any current task that is running for the provisioning scheme.

Metadata <Citrix.MachineCreation.Sdk.Metadata[]>

The metadata associated with this provisioning scheme.

ControllerAddress <string[]>

The DNS names of the controllers associated with this provisioning scheme for Quick Deploy purposes.

VMMetadata <char[]>

The opaque VM metadata block

UsePersonalVDiskStorage <bool>

True if the scheme will use personal vDisk storage.

PersonalVDiskDriveLetter <char>

The drive letter for the personal vDisk

PersonalVDiskDriveSize <int>

The size of the personal vDisk in GB

ProfileUsagePercentage <double>

The percentage of the personal vDisk to be used for profile data

DedicatedTenancy <bool>

Whether to use dedicated tenancy when creating machines in Cloud Hypervisors.

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### PartialData

Only a subset of the available data was returned.

#### CouldNotQueryDatabase

The query to get the database was not defined.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

#### CommunicationError

An error occurred while communicating with the service.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used, or examine the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

C:\PS>Get-ProvScheme

ProvisioningSchemeUid : 7585f0de-192e-4847-a6d8-22713c3a2f42

ProvisioningSchemeName : Scheme1

CpuCount : 1

MemoryMB : 1024

MasterImageVM : /Base.vm/base.snapshot

MasterImageVMDate : 17/05/2010 09:27:50

IdentityPoolUid : 03743136-e43b-4a87-af74-ab71686b3c16

IdentityPoolName : idPool1

HostingUnitUid : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501

HostingUnitName : HostUnit1

CleanOnBoot : True

TaskId : 00000000-0000-0000-0000-000000000000

Metadata : {Department = Sales}

ControllerAddress : {}

VMMetadata : {0, 1, 0, 0...}

ProvisioningSchemeUid : 43d82099-1fd7-4617-93f0-25b160813905

ProvisioningSchemeName : Scheme2

CpuCount : 1

MemoryMB : 1024

MasterImageVM : /Base.vm/base.snapshot

MasterImageVMDate : 17/05/2010 09:53:40

IdentityPoolUid : 03743136-e43b-4a87-af74-ab71686b3c16

IdentityPoolName : idPool1

HostingUnitUid : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501

HostingUnitName : HostUnit1

CleanOnBoot : True

TaskId : 00000000-0000-0000-0000-000000000000

Metadata : {}

ControllerAddress : {}  
VMMetadata : {0, 1, 0, 0...}

Returns all of the available provisioning schemes.

----- **EXAMPLE 2** -----

```
C:\PS>Get-ProvScheme -ProvisioningSchemeName Scheme[0-1]
```

ProvisioningSchemeUid : 7585f0de-192e-4847-a6d8-22713c3a2f42

ProvisioningSchemeName : Scheme1

CpuCount : 1

MemoryMB : 1024

MasterImageVM : /Base.vm/base.snapshot

MasterImageVMDate : 17/05/2010 09:27:50

IdentityPoolUid : 03743136-e43b-4a87-af74-ab71686b3c16

IdentityPoolName : idPool1

HostingUnitUid : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501

HostingUnitName : HostUnit1

CleanOnBoot : True

TaskId : 00000000-0000-0000-0000-000000000000

Metadata : {}

ControllerAddress : {}

VMMetadata : {0, 1, 0, 0...}

Returns all of the provisioning schemes that have the name 'Scheme0' or 'Scheme1'.

# Get-ProvSchemeMasterVMImageHistory

Jan 04, 2017

Gets the list of master VM snapshots that have been used to provide hard disks to provisioning schemes.

## Syntax

```
Get-ProvSchemeMasterVMImageHistory [-ProvisioningSchemeName <String>] [-ProvisioningSchemeUid <Guid>] [-MasterImageVM <String>] [-VMImageHistoryUid <Guid>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [-CommonParameters]
```

## Detailed Description

Provides the ability to discover the master VM snapshots that have been used previously to provide the hard disk image for provisioning schemes. This information includes the date and time when the image was introduced. This information can be used to roll back a provisioning scheme to a previous image.

## Related topics

[Publish-ProvMasterVMImage](#)

[Remove-ProvSchemeMasterVMImageHistory](#)

## Parameters

**-ProvisioningSchemeName**<String>

The name of the provisioning scheme.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ProvisioningSchemeUid**<Guid>

The unique identifier of the provisioning scheme.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-MasterImageVM**<String>

The path to the snapshot item in the hosting unit PowerShell provider.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-VMImageHistoryUid**<Guid>

The unique identifier for the Image History item.

Required?	false
-----------	-------



Default Value	
Accept Pipeline Input?	false

**-ReturnTotalRecordCount**<SwitchParameter>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-MaxRecordCount**<Int32>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-Skip**<Int32>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-Sort By**<String>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Filter**<String>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

Citrix.MachineCreation.Sdk.VMImage

The object that represents a master VM history. This contains the following parameters:

VMImageHistoryUid <Guid>

The unique identifier for the History item.

ProvisioningSchemeUid <Guid>

The unique identifier for the provisioning scheme that the VM was used for.

ProvisioningSchemeName <string>

The name of the provisioning scheme that the VM was used for.

MasterImageVM <string>

The path to the Snapshot item that was used as the master VM image.

Date <DateTime>

The date and time that the VM or snapshot was used in the provisioning scheme.

**Notes**

In the case of failure, the following errors can result.

**Error Codes**

-----

**PartialData**

Only a subset of the available data was returned.

**PermissionDenied**

The user does not have administrative rights to perform this operation.

**ConfigurationLoggingError**

The operation could not be performed because of a configuration logging error

**CouldNotQueryDatabase**

The query required to get the database was not defined.

**CommunicationError**

An error occurred while communicating with the service.

**DatabaseNotConfigured**

The operation could not be completed because the database for the service is not configured.

**InvalidFilter**

A filtering expression was supplied that could not be interpreted for this cmdlet.

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
C:\PS>Get-ProvSchemeMasterVMImageHistory
```

```
VMImageHistoryUid : 3cba3a75-89cd-4868-989b-27feb378fec5
ProvisioningSchemeUid : 7585f0de-192e-4847-a6d8-22713c3a2f42
ProvisioningSchemeName : MyScheme
MasterImageVM : /Base.vm/base.snapshot
Date : 17/05/2010 09:27:50
```

Gets all the hard disk images that have been used across all provisioning schemes.

----- **EXAMPLE 2** -----

```
C:\PS>Get-ProvSchemeMasterVMImageHistory -ProvisioningScheme MyScheme -masterImageVM "/BaseXp.vm/update1.snapshot" | Publish-ProvMasterVmImage
Roll back the provisioning scheme to use the hard disk from the update1.snapshot for the provisioning scheme called "MyScheme".
```

# Get-ProvScopedObject

Jan 04, 2017

Gets the details of the scoped objects for the MachineCreation Service.

## Syntax

```
Get-ProvScopedObject [-Scopeld <Guid>] [-ScopeName <String>] [-ObjectType <ScopedObjectType>] [-ObjectId <String>] [-ObjectName <String>] [-Description <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns a list of directly scoped objects including the names and identifiers of both the scope and object as well as the object description for display purposes.

There will be at least one result for every directly scoped object. When an object is associated with multiple scopes the output contains one result per scope duplicating the object details.

No records are returned for the All scope, though if an object is not in any scope a result with a null Scopeld and ScopeName will be returned.

## Related topics

### Parameters

#### **-Scopeld**<Guid>

Gets scoped object entries for the given scope identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

#### **-ScopeName**<String>

Gets scoped object entries with the given scope name.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ObjectType**<ScopedObjectType>

Gets scoped object entries for objects of the given type.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ObjectId**<String>

Gets scoped object entries for objects with the specified object identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ObjectName**<String>

Gets scoped object entries for objects with the specified object identifier.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-Description**<String>

Gets scoped object entries for objects with the specified description.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Prov\_Filtering for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

### **-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

### **-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Prov\_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

Citrix.MachineCreation.Sdk.ScopedObject

The Get-ProvScopedObject command returns an object containing the following properties:

ScopeId <Guid?>

Specifies the unique identifier of the scope.

ScopeName <String>

Specifies the display name of the scope.

ObjectType <ScopedObjectType>

Type of the object this entry relates to.

ObjectId <String>

Unique identifier of the object.

ObjectName <String>

Display name of the object

Description <String>

Description of the object (possibly \$null if the object type does not have a description).

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

UnknownObject

One of the specified objects was not found.

PartialData

Only a subset of the available data was returned.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

CouldNotQueryDatabase

The query required to get the database was not defined.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied



You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ProvScopedObject -ObjectType Scheme
```

```
ScopelId : eff6f464-f1ee-4442-add3-99982e0cec01
```

```
ScopeName : Sales
```

```
ObjectType : Scheme
```

```
ObjectId : cd4174ee-9e4b-4e57-b126-9dbf757fe493
```

```
ObjectName : MyExampleScheme
```

```
Description : Test scheme
```

```
ScopelId : 304e0fa7-d390-47f0-a94f-7e956a324c41
```

```
ScopeName : Finance
```

```
ObjectType : Scheme
```

```
ObjectId : cd4174ee-9e4b-4e57-b126-9dbf757fe493
```

```
ObjectName : MyExampleScheme
```

```
Description : Test scheme
```

```
ScopelId :
```

```
ScopeName :
```

```
ObjectType : Scheme
```

```
ObjectId : 5062e46b-71bc-4ac9-901a-30fe6797e2f6
```

```
ObjectName : AnotherScheme
```

```
Description : Another scheme in no scopes
```

Gets all of the scoped objects with type Scheme. The example output shows a scheme object (MyExampleScheme) in two scopes Sales and Finance, and another scheme (AnotherScheme) that is not in any scope. The ScopelId and ScopeName values returned are null in the final record.

# Get-ProvService

Jan 04, 2017

Gets the service record entries for the MachineCreation Service.

## Syntax

```
Get-ProvService [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns instances of the MachineCreation Service that the service publishes. The service records contain account security identifier information that can be used to remove each service from the database.

A database connection for the service is required to use this command.

## Related topics

### Parameters

#### **-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

#### **-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

#### **-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See `about_Prov_Filtering` for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by `-ReturnTotalRecordCount`.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-Sort By**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

### **-Filter**<String>

Gets records that match a PowerShell-style filter expression. See [about\\_Prov\\_Filtering](#) for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

Citrix.MachineCreation.Sdk.Service

The Get-ProvServiceInstance command returns an object containing the following properties.

Uid <Integer>

Specifies the unique identifier for the service in the group. The unique identifier is an index number.

ServiceHostId <Guid>

Specifies the unique identifier for the service instance.

DNSName <String>

Specifies the domain name of the host on which the service runs.

MachineName <String>

Specifies the short name of the host on which the service runs.

CurrentState <Citrix.Fma.Sdk.ServiceCore.ServiceState>

Specifies whether the service is running, started but inactive, stopped, or failed.

LastStartTime <DateTime>

Specifies the date and time at which the service was last restarted.

LastActivityTime <DateTime>

Specifies the date and time at which the service was last stopped or restarted.

OSType

Specifies the operating system installed on the host on which the service runs.

OSVersion

Specifies the version of the operating system installed on the host on which the service runs.

ServiceVersion

Specifies the version number of the service instance. The version number is a string that reflects the full build version of the service.

DatabaseUserName <string>

Specifies for the service instance the Active Directory account name with permissions to access the database. This will be either the machine account or, if the database is running on a controller, the NetworkService account.

Sid <string>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

ActiveSiteServices <string[]>

Specifies the names of active site services currently running in the service. Site services are components that perform long-running background processing in some services. This field is empty for services that do not contain site services.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### PartialData

Only a subset of the available data was returned.

#### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

#### CouldNotQueryDatabase

The query required to get the database was not defined.

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-ProvService
```

```
Uid          : 1
ServiceHostId : aef6f464-f1ee-4042-a523-66982e0cecd0
DNSName      : MyServer.company.com
MachineName  : MYSERVER
CurrentState  : On
LastStartTime : 04/04/2011 15:25:38
LastActivityTime : 04/04/2011 15:33:39
OSType       : Win32NT
OSVersion    : 6.1.7600.0
ServiceVersion : 5.1.0.0
DatabaseUserName : NT AUTHORITY\NETWORK SERVICE
SID          : S-1-5-21-2316621082-1546847349-2782505528-1165
ActiveSiteServices : {MySiteService1, MySiteService2...}
Get all the instances of the MachineCreation Service running in the current service group.
```

# Get-ProvServiceAddedCapability

Jan 04, 2017

Gets any added capabilities for the MachineCreation Service on the controller.

## Syntax

```
Get-ProvServiceAddedCapability [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables updates to the MachineCreation Service on the controller to be detected.

You do not need to configure a database connection before using this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

System.String

String containing added capabilities.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-ProvServiceAddedCapability
```

Get the added capabilities of the MachineCreation Service.



# Get-ProvServiceConfigurationData

Jan 04, 2017

Gets configuration data for the service.

## Syntax

```
Get-ProvServiceConfigurationData [-Name <String[]>] [-Value <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to determine the configuration parameters for the service

## Related topics

[Set-ProvServiceConfigurationData](#)

[Remove-ProvServiceConfigurationData](#)

## Parameters

**-Name**<String[]>

The Configuration data item Name .

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Value**<String[]>

The Configuration data item value.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Prov\\_Filtering](#) for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

**-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Prov\_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

Citrix.MachineCreation.Sdk.ServiceConfigurationData

This object provides details of the configuration data and contains the following information:

**Name** <string>

The Name for the configuration item.

**Value** <string>

The value for the configuration item.

**Notes**

In the case of failure the following errors can be produced.

**Error Codes**

-----

**PartialData**

Only a subset of the available data was returned.

**CouldNotQueryDatabase**

The query required to get the database was not defined.

**DatabaseError**

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

An error occurred while communicating with the service.

ExceptionThrown

An unexpected error occurred. To locate more details see the Windows event logs on the controller being used, or examine the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
C:\PS>Get-ProvConfiguratuionData
```

```
Name      : DeltaDiskDelete.timeDelay
```

```
Value     : 10
```

Get the Configuration data for the service.

# Get-ProvServiceInstance

Jan 04, 2017

Gets the service instance entries for the MachineCreation Service.

## Syntax

```
Get-ProvServiceInstance [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns service interfaces published by the instance of the MachineCreation Service. Each instance of a service publishes multiple interfaces with distinct interface types, and each of these interfaces is represented as a ServiceInstance object. Service instances can be used to register the service with a central configuration service so that other services can use the functionality.

You do not need to configure a database connection to use this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.MachineCreation.Sdk.ServiceInstance

The Get-ProvServiceInstance command returns an object containing the following properties.

ServiceGroupUid <Guid>

Specifies the unique identifier for the service group of which the service is a member.

ServiceGroupName <String>

Specifies the name of the service group of which the service is a member.

ServiceInstanceUID <Guid>

Specifies the unique identifier for registered service instances, which are service instances held by and obtained from a

central configuration service. Unregistered service instances do not have unique identifiers.

ServiceType <String>

Specifies the service instance type. For this service, the service instance type is always Prov.

Address

Specifies the address of the service instance. The address can be used to access the service and, when registered in the central configuration service, can be used by other services to access the service.

Binding

Specifies the binding type that must be used to communicate with the service instance. In this release of XenDesktop, the binding type is always 'wcf\_HTTP\_kerb'. This indicates that the service provides a Windows Communication Foundation endpoint that uses HTTP binding with integrated authentication.

Version

Specifies the version of the service instance. The version number is used to ensure that the correct versions of the services are used for communications.

ServiceAccount <String>

Specifies the Active Directory account name for the machine on which the service instance is running. The account name is used to provide information about the permissions required for interservice communications.

ServiceAccountSid <String>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

InterfaceType <String>

Specifies the interface type. Each service can provide multiple service instances, each for a different purpose, and the interface defines the purpose. Available interfaces are:

SDK - for PowerShell operations

InterService - for operations between different services

Peer - for communications between services of the same type

Metadata <Citrix.MachineCreation.Sdk.Metadata[]>

The collection of metadata associated with registered service instances, which are service instances held by and obtained from a central configuration service. Metadata is not stored for unregistered service instances.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-ProvServiceInstance
```

```
Address      : http://MyServer.com:80/Citrix/MachineCreationService
Binding      : wcf_HTTP_kerb
InterfaceType : SDK
Metadata     :
MetadataMap  :
ServiceAccount : ENG\MyAccount$
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164
ServiceGroupName : MyServiceGroup
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000
ServiceType   : Prov
Version      : 1
```

```
Address      : http://MyServer.com:80/Citrix/MachineCreationService/IServiceApi
Binding      : wcf_HTTP_kerb
InterfaceType : InterService
Metadata     :
MetadataMap  :
```

ServiceAccount : ENG\MyAccount  
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164  
ServiceGroupName : MyServiceGroup  
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d  
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000  
ServiceType : Prov  
Version : 1

Get all instances of the MachineCreation Service running on the specified machine. For remote services, use the AdminAddress parameter to define the service for which the interfaces are required. If the AdminAddress parameter has not been specified for the runspace, service instances running on the local machine are returned.



# Get-ProvServiceStatus

Jan 04, 2017

Gets the current status of the MachineCreation Service on the controller.

## Syntax

```
Get-ProvServiceStatus [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables the status of the MachineCreation Service on the controller to be monitored. If the service has multiple data stores it will return the overall state as an aggregate of all the data store states. For example, if the site data store status is OK and the secondary data store status is DBUnconfigured then it will return DBUnconfigured.

## Related topics

[Set-ProvDBConnection](#)

[Test-ProvDBConnection](#)

[Get-ProvDBConnection](#)

[Get-ProvDBSchema](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Get-ProvServiceStatus command returns an object containing the status of the MachineCreation Service together with extra diagnostics information.

DBUnconfigured

The MachineCreation Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the MachineCreation Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

#### InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the MachineCreation Service schema has not been added to the database.

#### DBNotFound

The specified database could not be located with the configured connection string.

#### DBNewerVersionThanService

The version of the MachineCreation Service currently in use is incompatible with the version of the MachineCreation Service schema on the database. Upgrade the MachineCreation Service to a more recent version.

#### DBOlderVersionThanService

The version of the MachineCreation Service schema on the database is incompatible with the version of the MachineCreation Service currently in use. Upgrade the database schema to a more recent version.

#### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

#### OK

The MachineCreation Service is running and is connected to a database containing a valid schema.

#### Failed

The MachineCreation Service has failed.

#### Unknown

(0) The service status cannot be determined.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ProvServiceStatus
```

DBUnconfigured

Get the current status of the MachineCreation Service.

# Get-ProvTask

Jan 04, 2017

Gets the task history for the MachineCreation Service.

## Syntax

```
Get-ProvTask [-TaskId] <Guid> [-Type <JobType>] [-Active <Boolean>] [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns a list of tasks that have run or are currently running within the MachineCreation Service.

## Related topics

[Remove-ProvTask](#)

[Stop-ProvTask](#)

[Switch-ProvTask](#)

[Add-ProvTaskMetadata](#)

[Remove-ProvTaskMetadata](#)

## Parameters

**-TaskId**<Guid>

Specifies the task identifier to be returned.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Type**<JobType>

Specifies the type of task to be returned.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Active**<Boolean>

Specifies whether currently running tasks only or completed tasks only are returned.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: -Metadata "abc:x\*" matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through Select-Object, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See about\_Prov\_Filtering for details.

Required?	false
Default Value	False

Accept Pipeline Input?	false
------------------------	-------

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

**-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Prov\_Filtering for details.

Required?	false
-----------	-------

Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can result.

Error Codes

-----

UnknownObject

One of the specified objects was not found.

PartialData

Only a subset of the available data was returned.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

CouldNotQueryDatabase

The query required to get the database was not defined.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for

various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ProvTask -Active $false
```

```
TaskId           : cfe5b3a2-c471-443e-b05e-8658e672d10f
Type             : MyTask
Host             : NYSERVER
Status          : Finished
CurrentOperation :
TaskProgress     : 100
TaskExpectedCompletion : 10/10/2012 15:28:12
LastUpdateTime  : 10/10/2012 15:28:12
ActiveElapsedTime : 56
DateFinished    : 10/10/2012 15:28:12
TerminatingError :
```

...

Get all completed tasks for the MachineCreation Service.

All tasks will publish at least the fields listed above, plus more related to the particular task being performed.



# Get-ProvVM

Jan 04, 2017

Gets the VMs that were created using Citrix XenDesktop Machine Creation Services.

## Syntax

```
Get-ProvVM [-ProvisioningSchemeName] <String> [-ProvisioningSchemeUid <Guid>] [-VMName <String>] [-Locked <Boolean>] [-Tag <String>] [-OutOfDate] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to obtain a list of the VMs that were created using Machine Creation Services.

## Related topics

[New-ProvVM](#)

[Remove-ProvVM](#)

[Lock-ProvVM](#)

[Unlock-ProvVM](#)

## Parameters

**-ProvisioningSchemeName**<String>

The name of the provisioning scheme.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ProvisioningSchemeUid**<Guid>

The unique identifier of the provisioning scheme.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-VMName**<String>

The name of the VM in the hypervisor context.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Locked**<Boolean>

Indicates whether only VMs that are marked as locked are returned or not (see Lock-ProvVM and Unlock-ProvVM for details).

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Tag**<String>

The tag string that was associated with the VM when it was locked.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-OutOfDate**<SwitchParameter>

Indicates that the image currently assigned to the VM is out of date. The image will be updated the next time the VM is restarted.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ReturnTotalRecordCount**<SwitchParameter>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-MaxRecordCount**<Int32>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-Skip**<Int32>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-SortBy**<String>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Filter**<String>

See about\_Prov\_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

#### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

#### Return Values

Citrix.MachineCreation.Sdk.ProvisionedVirtualMachine

The object has the following properties:

ADAccountSid <string>

The SID of the AD computer account that the VM is using.

ADAccountName <string>

The name of the AD computer account that the VM is using.

CpuCount <int>

The number of processors that the VM has been allocated.

CreationDate <DateTime>

The date and time that the VM was created.

Domain <string>

The Domain of the AD computer account that the VM is using.

ImageOutOfDate <Boolean>

Indicates if the image will be updated next time the VM is started.

Lock <Boolean>

Indicates whether the VM is locked or not.

MemoryMB <int>

The maximum amount of memory that VMs will be created with when using this scheme.

ProvisioningSchemeName <string>

The name of the provisioning scheme that the VM is part of.

ProvisioningSchemeUid <Guid>

The unique identifier for the provisioning scheme that the VM is part of.

Tag <string>

Provides the string associated with a locked VM.

VMId <string>

The identifier for the VM (hypervisor context).

VMName <string>

The name of the VM (hypervisor context).

AssignedImage <string>

The identifier (in the hypervisor) for the hard disk image that the VM is currently assigned.

BootedImage <string>

The identifier (in the hypervisor) for the hard disk image with which the VM is currently started.

HostingUnitUid <Guid>

The unique identifier for the hosting unit that was used to create the VM.

HypervisorConnectionUid <Guid>

The unique identifier of the hypervisor connection that was used to create the VM.

LastBootTime <DateTime>

The date and time of the last start of the VM.

OSDiskIndex <int>

The disk index at which the hard disk image, from which the VM is currently started, is attached (or [int]::MinValue for VMs inherited from versions of XenDesktop before 5.6)

PersonalVDiskIndex <int>

The disk index at which the personal vdisk is attached (defaults to [int]::MinValue for VMs without a personal vdisk)

PersonalVDiskStorage <string>

The identifier (in the hypervisor) for the storage on which the personal virtual disk image from which the VM is currently started is located. This is set only if the VM has a personal vDisk attached

StorageId <string>

The identifier (in the hypervisor) for the storage on which the hard disk image, from which the VM is currently started, is located.

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### PartialData

Only a subset of the available data was returned.

#### CouldNotQueryDatabase

The query required to get the database was not defined.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

#### CommunicationError

An error occurred while communicating with the service.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS>Get-ProvVM -provisioningSchemeName MyScheme
```

```

ADAccountName      : MYDomain\computer2$
ADAccountSid       : S-1-5-21-3751941309-1176885247-1409628468-3179
CpuCount           : 1
CreationDate       : 17/05/2012 09:35:22
Domain             : steve.dum.local
ImageOutOfDate    : False
Lock               : True
MemoryMB           : 512
ProvisioningSchemeName : XenPS
ProvisioningSchemeUid : 5135a865-ba49-4e5f-87f2-2d65ee7a4e51
Tag                : Brokered
VMId               : a830de93-ddc5-b763-dc1a-35580a31401c
VMName            : IP0051
AssignedImage      : 57fc60c3-eb9b-4d38-8646-0afceec85335
BootedImage        : 57fc60c3-eb9b-4d38-8646-0afceec85335
HostingUnitUid     : ea17840f-cf2d-4d80-94e0-3b752b32e0af
HypervisorConnectionUid : 99f9f826-31fc-4453-8ca0-9ba54306c3ac
IdentityDiskIndex : 1
LastBootTime       : 17/05/2012 09:35:22
OSDiskIndex        : 0
PersonalVDiskIndex : -2147483648
PersonalVDiskStorage :
StorageId          : 33ad07a7-edd7-589b-716a-86cad4739f5e
Gets all the Virtual Machines that were provisioned into the Provisioning Scheme called 'MyScheme'.

```

----- **EXAMPLE 2** -----

```
C:\PS>Get-ProvVM -Locked $true
```

```

ADAccountName      : MYDomain\computer1$
ADAccountSid       : S-1-5-21-3751941309-1176885247-1409628468-3178
CpuCount           : 1
CreationDate       : 17/05/2012 09:35:30
Domain             : steve.dum.local
ImageOutOfDate    : False
Lock               : True
MemoryMB           : 512
ProvisioningSchemeName : XenPS
ProvisioningSchemeUid : 5135a865-ba49-4e5f-87f2-2d65ee7a4e51
Tag                : Brokered
VMId               : a830de93-ddc5-b763-dc1a-35580a31401c
VMName            : IP0051
AssignedImage      : 57fc60c3-eb9b-4d38-8646-0afceec85335
BootedImage        : 57fc60c3-eb9b-4d38-8646-0afceec85335
HostingUnitUid     : ea17840f-cf2d-4d80-94e0-3b752b32e0af
HypervisorConnectionUid : 99f9f826-31fc-4453-8ca0-9ba54306c3ac
IdentityDiskIndex : 1
LastBootTime       : 17/05/2012 09:35:22

```

OSDiskIndex : 0

PersonalVDiskIndex : -2147483648

PersonalVDiskStorage :

StorageId : 33ad07a7-edd7-589b-716a-86cad4739f5e

Gets all the Virtual Machines that were locked, regardless of which Provisioning Scheme the VM is part of.



# Lock-ProvVM

Jan 04, 2017

Locks a VM.

## Syntax

```
Lock-ProvVM [-VMID] <String[]> -ProvisioningSchemeName <String> [-Tag <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Lock-ProvVM [-VMID] <String[]> -ProvisioningSchemeUid <Guid> [-Tag <String>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to 'lock' a virtual machine with a tag string. This stops other commands from being able to remove the virtual machine without unlocking the VM first. This is to enable consumers of the virtual machines to indicate that the VM is being used.

## Related topics

[UnLock-ProvVM](#)

[Get-ProvVM](#)

[Remove-ProvVM](#)

## Parameters

**-VMID**<String[]>

The virtual machine Id (hypervisor context).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ProvisioningSchemeName**<String>

The name of the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ProvisioningSchemeUid**<Guid>

The unique identifier of the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Tag**<String>

The string to be held against the VM being locked.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Direct or typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Input Type

Citrix.MachineCreation.Sdk.ProvisionedVirtualMachine You can pipe an object containing a parameter called 'VMId' and 'ProvisioningSchemeName' to Lock-ProvVM

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### VMDoesNotExist

The specified VM cannot be located.

#### VMAlreadyLocked

The VM is already unlocked.

#### VMDoesNotExistForProvisioningScheme

The specified VM does exist in the hypervisor, but is not part of the specified provisioning scheme.

#### PermissionDenied

The user is not authorized to perform this operation

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the

XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS>Lock-ProvVM -provisioningSchemeName MyScheme -VMId bc79802c-ba6e-8de8-99e9-4c35d7ad24b4 -Tag LockedString
```

Locks the VM with the Id 'bc79802c-ba6e-8de8-99e9-4c35d7ad24b4' in the provisioning scheme 'MyScheme' with the tag 'LockedString'.

### ----- EXAMPLE 2 -----

```
C:\PS>Get-ProvVM -provisioningSchemeName MyScheme | Lock-ProvVM -Tag LockedString
```

Locks all the VMs in the provisioning scheme 'MyScheme' with the tag 'LockedString'.

# New-ProvScheme

Jan 04, 2017

Creates a new provisioning scheme.

## Syntax

```
New-ProvScheme [-ProvisioningSchemeName] <String> -HostingUnitName <String> -IdentityPoolName <String> -MasterImageVM <String> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-CleanOnBoot] [-UsePersonalVDiskStorage] [-Scope <String[]>] [-NoImagePreparation] [-NetworkMapping <Hashtable>] [-Metadata <Hashtable>] [-ServiceOffering <String>] [-SecurityGroup <String[]>] [-DedicatedTenancy] [-VhdTemplateSource <String>] [-VhdResultDestination <String>] [-AppScanResultsFile <String>] [-ResetAdministratorPasswords] [-RunAsynchronously] [-PurgeJobOnSuccess] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-ProvScheme [-ProvisioningSchemeName] <String> -HostingUnitUid <Guid> -IdentityPoolUid <Guid> -MasterImageVM <String> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-CleanOnBoot] [-UsePersonalVDiskStorage] [-Scope <String[]>] [-NoImagePreparation] [-NetworkMapping <Hashtable>] [-Metadata <Hashtable>] [-ServiceOffering <String>] [-SecurityGroup <String[]>] [-DedicatedTenancy] [-VhdTemplateSource <String>] [-VhdResultDestination <String>] [-AppScanResultsFile <String>] [-ResetAdministratorPasswords] [-RunAsynchronously] [-PurgeJobOnSuccess] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Lets you create a new provisioning scheme. The creation process makes a copy of the hard disk attached to a virtual machine snapshot and stores it in every storage location that the hosting unit referenced by the provisioning scheme defines. This is a long-running task and typically takes several minutes to complete (depending on the size of the hard disk that is being copied and the number of snapshots that the hard disk consists of).

A snapshot must be used rather than a VM, so that the content of the hard disk for the provisioning scheme can be easily determined.

As the snapshot is specified using a path into the Citrix Host Service Powershell Provider the Citrix Host Service Powershell snap-in must also be loaded for this cmdlet to be used.

This cmdlet requires information to be provided that is retrieved using other snap-ins that form part of the Citrix Machine Creation Services: Hosting Unit Service Snapin The snap-in that provides information about the hypervisors. AD Identity Service Snapin The snap-in that provides information about the identity pools.

The provisioning scheme is a collection of all of the data that is required to form a template against which virtual machines can be created. It therefore requires the following: Hosting Unit A reference to an item defined in the Host Service that defines the hypervisor, the network, and the storage to be used. Identity Pool A reference to the collection of Active Directory accounts that is used for virtual machines created from the provisioning scheme.

## Related topics

[Get-ProvTask](#)

[Get-ProvScheme](#)

[Test-ProvSchemeNameAvailable](#)

## Parameters

**-ProvisioningSchemeName**<String>

The name of the provisioning scheme to be created. This must be a name that is not being used by an existing provisioning scheme, and it must not contain any of the following characters \;#.\*?=<>|[]0""

Required?	true
Default Value	
Accept Pipeline Input?	false

**-HostingUnitName**<String>

The name of the hosting unit used for the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-IdentityPoolName**<String>

The name for the identity pool used for the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-HostingUnitUid<Guid>**

The identifier for the hosting unit used for the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-IdentityPoolUid<Guid>**

The identifier of the identity pool used for the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-MasterImageVM<String>**

The path in the hosting unit provider to the virtual machine snapshot that is used. This identifies the hard disk to be used and the default values for the memory and processors. This must be a path to a Snapshot item in the same hosting unit that the hosting unit name or hosting unit UID refers to.

Valid paths are of the format; XDHyp:\HostingUnits\<HostingUnitName>\<path>\<VmName>.vm\<SnapshotName>.snapshot

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-VMCpuCount<Int32>**

The number of processors used by virtual machines created from the provisioning scheme.

Required?	false
Default Value	The number of CPUs assigned to the base image VM snapshot.
Accept Pipeline Input?	false

**-VMMemoryMB<Int32>**

The maximum amount of memory used by virtual machines created from the provisioning scheme.

Required?	false
Default Value	The amount of memory assigned to the base image VM snapshot.
Accept Pipeline Input?	false

**-CleanOnBoot<SwitchParameter>**

Indicates whether or not the virtual machines created from this provisioning scheme are reset to their initial condition each time they are started.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-UsePersonalVDiskStorage<SwitchParameter>**

Indicates whether or not personal vDisks are used for the VMs in this provisioning scheme.

Required?	false
-----------	-------

Default Value	
Accept Pipeline Input?	false

**-Scope**<String[]>

The administration scopes to be applied to the new provisioning scheme.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-NoImagePreparation**<SwitchParameter>

Indicates that Image Preparation should not be performed on this Provisioning Scheme

Required?	false
Default Value	
Accept Pipeline Input?	false

**-NetworkMapping**<Hashtable>

Specifies how the attached NICs are mapped to networks. If this parameter is omitted, provisioned VMs are created with a single NIC, which is mapped to the default network in the HostingUnit. If this parameter is supplied, machines are created with the number of NICs specified in the map, and each NIC is attached to the specified network.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Metadata**<Hashtable>

Specifies any metadata to be attached to the scheme when it is created.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ServiceOffering**<String>

The Service Offering to use when creating machines in Cloud Hypervisors.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-SecurityGroup**<String[]>

The security groups to apply to machines created in Cloud Hypervisors

Required?	false
Default Value	
Accept Pipeline Input?	false

**-DedicatedTenancy**<SwitchParameter>

Whether to use dedicated tenancy when creating machines in Cloud Hypervisors.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-VhdTemplateSource**<String>

A file path to a source VHD template to be used when performing Application Scanning during Image Preparation. The presence of this parameter in conjunction with VhdResultDestination implies that application scanning is to be performed

Required?	false
Default Value	
Accept Pipeline Input?	false

**-VhdResultDestination**<String>

A file path (including file name) where the VHD disk file containing the results of application scanning should be written.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AppScanResultsFile**<String>

File name to which the results of application scanning should be written.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ResetAdministratorPasswords**<SwitchParameter>

Indicates whether to reset the password for the administrator accounts on provisioned machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-RunAsynchronously**<SwitchParameter>

Indicates whether or not the command returns before it is complete. If specified, the command returns an identifier for the task that was created. This task can be monitored using the get-ProvTask command.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-PurgeJobOnSuccess**<SwitchParameter>

Indicates that the task history is removed from the database when the task has finished. This can only be specified for tasks that are not run asynchronously.

Required?	false
-----------	-------



Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

System.Guid

When the RunAsynchronously identifier is specified, this GUID is returned and provides the task identifier.

System.Management.Automation.PSCustomObject

This object provides details of the task that was run and contains the following information:

**TaskId** <Guid>

The identifier for the task that was performed.

**Active** <Boolean>

Indicates whether the task is still processing or is complete.

**Host** <string>

The name of the host on which the task is running or was run.

**DateStarted** <DateTime>

The date and time that the task was initiated.

**Type** <Citrix.XDInterServiceTypes.JobType>

The type of task. For new provisioning scheme tasks, this is always NewProvisioningScheme.

**Metadata** <Citrix.MachineCreation.Sdk.Metadata[]>

The list of metadata stored against the task. For new tasks, this is empty until metadata is added.

**WorkflowStatus** <System.Workflow.Runtime.WorkflowStatus>

Indicates the status of the workflow that is used to process the task.

**ProvisioningSchemeName** <string>

The name of the provisioning scheme that the task was for.

**ProvisioningSchemeUid** <Guid>

The unique identifier of the provisioning scheme that the task was for.

**MasterImage** <string>

The path (in the hosting unit provider) of the virtual machine snapshot that was used as the master VM image for the task.

**IdentityPoolName** <string>

The name of the identity pool (from the ADIdentity PowerShell snap-in) that the new provisioning scheme uses.

**IdentityPoolUid** <guid>

The unique identifier name of the identity pool (from the ADIdentity PowerShell snap-in) that the new provisioning scheme uses.

HostingUnitName <string>

The name of the hosting unit (from the Hosting Unit PowerShell snap-in) that the new provisioning scheme uses.

HostingUnitUid

The unique identifier of the hosting unit (from the Hosting Unit PowerShell snap-in) that the new provisioning scheme uses.

PersonalVdiskDriveLetter

The drive letter on which a personal vDisk is mounted (blank if the personal vDisk feature was not selected).

PersonalVdiskDriveSize

The size of any personal vDisk (zero if the personal vDisk feature was not selected).

Scopes <Citrix.Fma.Sdk.ServiceCoreScopeReference[]>

The delegated administration scopes to which the scheme will belong.

NetworkMap <Citrix.MachineCreation.Sdk.NetworkMap>

The list of NIC to network associations, if specified.

ProvisioningSchemeMetadata <Dictionary<string, string>>

The metadata to apply to the provisioning scheme, if specified.

TaskState <Citrix.MachineCreation.Sdk.NewProvisioningSchemeState>

The state of the task. This can be any of the following:

Processing

The task has begun but has not done anything yet.

LocatingResources,

The workflow is locating resources from other services.

HostingUnitNotFound

The task failed because the required hosting unit could not be located.

VirtualMachineSnapshotNotFound

The task failed because the required VM snapshot could not be located.

ConsolidatingMasterImage

The task is consolidating the master image.

ReplicatingConsolidatedImageToAllStorage

The task is replicating the consolidated master image.

StoringProvisioningScheme

The task is storing the provisioning scheme data in the database.

Finished

The task completed with no errors.

ProvisioningSchemeAlreadyExists

The task failed because a provisioning scheme with the same name already exists.

IdentityPoolNotFound

The task failed because the specified identity pool could not be found.

MasterVMImageIsNotPartOfProvisioningSchemeHostingUnit,

The task failed because the hosting unit from which the master image originated is not the same hosting unit that the provisioning scheme is using.

MasterVmlmageIsASnapshot

The task failed because the master VM path does not refer to a Snapshot item.

ProvisioningSchemeNotFound

The task failed because it could not find a provisioning scheme with the specified name.

TaskAlreadyRunningForProvisioningScheme

The task failed because a task for a provisioning scheme with the same name is already running.

InvalidMasterVMConfiguration

The task failed because the VM snapshot specified as the master has an invalid configuration.

#### InvalidMasterVMState

The task failed because the VM snapshot specified as the master is currently in an invalid state.

#### InsufficientResources

The task failed because the hypervisor did not have enough resources to complete the task.

#### DiskConsolidationFailed

The disk consolidation task failed. Details are in the task state information string.

#### StorageNotFound

The task failed because no associated storage was found in the hosting unit.

#### ConfigurationError

The task failed because the service is unable to contact one of the other services. This is because not all appropriate Configuration Service registrations have been performed.

#### Canceled

The task was stopped by user intervention (using Stop-ProvTask).

#### TaskStateInformation

Additional information about the current task state.

#### TaskProgress

The progress of the task 0-100%.

#### DiskSize

The size of the master image in GB

#### DedicatedTenancy

Whether to use dedicated tenancy when creating machines in Cloud Hypervisors.

#### Notes

Only one long-running task for each provisioning scheme can be processed at a time.

In case of failure, the following errors can result.

#### Error Codes

-----

#### JobCreationFailed

The requested task could not be started.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation. Communication with the database failed for various reasons.

#### MachineCreationServiceDoesNotSupportPersonalDisk

The service instance being used has not been upgraded to support the personal vDisk feature.

#### DatabaseMissingCapabilities

The database supporting the service instance being used has not been upgraded to support the personal vDisk feature.

#### CommunicationError

An error occurred while communicating with the service.

#### InvalidParameterCombination

Both PurgeJobOnSuccess and RunAsynchronously were specified. When running asynchronously, the cmdlet terminates before the job does, so it cannot clean up the completed job.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### ScopeNotFound

One or more of the scopes nominated for the new provisioning scheme do not exist.

#### WorkflowHostUnavailable

The task could not be started because the database connection is inactive.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used, or examine the XenDesktop logs. VhdParametersMustBeSupplied

When parameter VhdTemplateSource or VhdResultDestination is supplied, both parameters are required to be supplied.

The cmdlet is associated with a task of type NewProvisioningScheme, and while active will move through the following operations (CurrentOperation field)

#### ValidatingInputs

#### ConsolidatingMasterImage

#### PreparingMasterImage

#### ReplicatingMasterImage

#### CommittingScheme

#### Examples

##### ----- EXAMPLE 1 -----

```
C:\PS> New-ProvScheme -ProvisioningSchemeName XenPS -HostingUnitName XenHu -IdentityPool  
IName idPool1 -CleanOnBoot -MasterImageVM XDHyp:\HostingUnits\XenHU\Base.vm\Base.snapshot
```

```
TaskId           : 90e93b9d-a225-4701-ad50-fa1546af35ac  
Active           : False  
Host             : MyHost  
DateStarted      : 17/05/2010 08:22:22  
Type             : NewProvisioningScheme  
Metadata         : {}  
WorkflowStatus   : Completed  
ProvisioningSchemeName : XenPS  
MasterImage      : /Base.vm/Base.snapshot  
IdentityPoolName : idPool1  
IdentityPoolUid  : 03743136-e43b-4a87-af74-ab71686b3c16  
HostingUnitName  : XenHU  
HostingUnitUid   : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501  
PersonalVDiskDriveLetter :  
PersonalVDiskDriveSize : 0  
ProvisioningSchemeUid : 7585f0de-192e-4847-a6d8-22713c3a2f42  
CurrentOperation :  
TaskState        : Finished  
TaskStateInformation :  
TaskProgress     : 100  
DiskSize         : 24  
Creates a new provisioning scheme with the name "XenPS" using the hosting unit "XenHu" and the identity pool "idPool1" from the master VM snapshot called "Base.snapshot".
```

##### ----- EXAMPLE 2 -----

```
C:\PS> New-ProvScheme -ProvisioningSchemeName XenPS -HostingUnitName XenHu -IdentityPoolName idPool1 -CleanOnBoot -MasterImageVM XDHyp:\HostingUnits\XenHU\Base.vm\Base.snapst
```

#### Guid

```
-----  
6dd85fec-96cf-46b1-9cd4-d8ba7d06e85b
```

Creates a new provisioning scheme with the name "XenPS" using the hosting unit "XenHu" and the identity pool "idPool1" from the master VM snapshot called "Base.snapshot" asynchronously. To get the task details, use Get-ProvTask -TaskID <task id>

i.e.

```
C:\PS> Get-ProvTask -TaskID 6dd85fec-96cf-46b1-9cd4-d8ba7d06e85b
```

```
TaskId : 6dd85fec-96cf-46b1-9cd4-d8ba7d06e85b
```

```
Active : False
```

```
Host : MyHost
```

```
DateStarted : 17/05/2010 08:22:22
```

```
Type : NewProvisioningScheme
```

```
Metadata : {}
```

```
WorkflowStatus : Completed
ProvisioningSchemeName : XenPS
MasterImage : XDHyp:\HostingUnits\XenHU\Base.vm\Base.snapshot
IdentityPoolName : idPool1
IdentityPoolUid : 03743136-e43b-4a87-af74-ab71686b3c16
HostingUnitName : XenHU
HostingUnitUid : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501
PersonalVdiskDriveLetter :
PersonalVdiskDriveSize : 0
ProvisioningSchemeUid : 7585f0de-192e-4847-a6d8-22713c3a2f42
TaskState : Finished
TaskStateInformation :
TaskProgress : 100
DiskSize : 24
```

----- **EXAMPLE 3** -----

```
C:\PS>$provScheme = New-ProvScheme -ProvisioningSchemeName XenPS2 -HostingUnitName XenHu -IdentityPoolName idPool1 -CleanOnBoot -MasterImageVM XDHyp:\HostingUnits\XenHU\Base
Creates a new provisioning scheme with the name "XenPS2" using the hosting unit "XenHu" and the identity pool "idPool1" from the master VM snapshot called "Base.snapshot"; apply a 17GB
personal vDisk. The personal vDisk is mapped as drive X. The operation runs synchronously, and the return value contains the task details
```

For example:

```
C:\PS>$provScheme
TaskId : d726222a-04b5-4098-b9ac-d85ed9d351b
Active : False
Host : MyHost
DateStarted : 12/09/2011 09:30:04
Type : NewProvisioningScheme
Metadata : {}
ProvisioningSchemeName : XenPS2
IdentityPoolName : idPool1
IdentityPoolUid : 03743136-e43b-4a87-af74-ab71686b3c16
HostingUnitName : XenHU
HostingUnitUid : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501
PersonalVdiskDriveLetter : X
PersonalVdiskDriveSize : 17
WorkflowStatus : Completed
MasterImage : XDHyp:\HostingUnits\XenHU\Base.vm\Base.snapshot
ProvisioningSchemeUid : 7585f0de-192e-4847-a6d8-22713c3a2f42
TaskState : Finished
TaskStateInformation :
TaskProgress : 100
DiskSize : 24
```

# New-ProvVM

Jan 04, 2017

Creates a new virtual machine.

## Syntax

```
New-ProvVM -ProvisioningSchemeName <String> -ADAccountName <String[]> [-FastBuild] [-NetworkMapping <Hashtable>] [-AutoAssignVLAN] [-SecurityGroup <String[]>] [-MachinesPerAssistant <Int32>] [-MaxAssistants <Int32>] [-RunAsynchronously] [-PurgeJobOnSuccess] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
New-ProvVM -ProvisioningSchemeUid <Guid> -ADAccountName <String[]> [-FastBuild] [-NetworkMapping <Hashtable>] [-AutoAssignVLAN] [-SecurityGroup <String[]>] [-MachinesPerAssistant <Int32>] [-MaxAssistants <Int32>] [-RunAsynchronously] [-PurgeJobOnSuccess] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Lets you create new virtual machines with the configuration specified by a provisioning scheme.

The virtual machines are created using all of the storage specified in the provisioning scheme. The storage is used in a round robin mechanism. For the duration of this task, the AD accounts are locked (by the AD Identity Service), which stops the same accounts being used by other virtual machine creation tasks.

## Related topics

[Get-ProvVM](#)

[Remove-ProvVM](#)

[Lock-ProvVM](#)

[Unlock-ProvVM](#)

## Parameters

**-ProvisioningSchemeName**<String>

The name of the provisioning scheme in which the virtual machines are created.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-ProvisioningSchemeUid**<Guid>

The unique identifier for the provisioning scheme in which the virtual machines are created.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-ADAccountName**<String[]>

A list of the Active Directory account names that are used for the virtual machines. The accounts must be provided in a domain-qualified format. This parameter accepts Identity objects as returned by the New-AcctADAccount cmdlet, or any PSObject with string properties "Domain" and "ADAccountName".

Required?	true
Default Value	
Accept Pipeline Input?	false

**-FastBuild**<SwitchParameter>

Indicates whether or not the command can improve speed by using optimizations in the creation process. This may mean that machines you create cannot be booted until ResetVM operations have been performed by the Broker and Machine Identity Services.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-NetworkMapping**<Hashtable>

Specifies how the attached NICs are mapped to networks. If this parameter is omitted, provisioned VMs are created with network settings as inherited from the provisioning scheme. If this parameter is supplied, machines are created with the number of NICs specified in the map, and each NIC is attached to the specified network.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AutoAssignVLAN**<SwitchParameter>

Indicates whether or not the VLAN of the network should be automatically assigned to the new VM or if the VLAN from the master image should be used

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-SecurityGroup**<String[]>

The security groups to apply to machines created in Cloud Hypervisors

Required?	false
Default Value	
Accept Pipeline Input?	false

**-MachinesPerAssistant**<Int32>

The number of concurrent volume operations that may be performed by each volume worker helper machine

Required?	false
Default Value	
Accept Pipeline Input?	false

**-MaxAssistants**<Int32>

The maximum number of volume worker help machines that may be created for this operation.

Required?	false
Default Value	
Accept Pipeline Input?	false



**-RunAsynchronously**<SwitchParameter>

Indicates whether or not the command returns before it is complete. If specified, the command returns an identifier for the task that was created. This task can be monitored using the get-ProvTask command.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-PurgeJobOnSuccess**<SwitchParameter>

Indicates that the task history is removed from the database when the task has finished. This cannot be specified for tasks that are run asynchronously.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller to which the PowerShell snap-in connects. You can provide this as a host name or an IP address.

Required?	false
Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.

Accept Pipeline Input?	false
------------------------	-------

## Return Values

### System.Guid

When the RunAsynchronously identifier is specified, this GUID is returned and provides the task identifier.

### System.Management.Automation.PSCustomObject

This object provides details of the task that was run and contains the following information:

#### TaskId <Guid>

The identifier for the task that was performed.

#### Active <Boolean>

Indicates whether the task is still processing or is complete.

#### Host <string>

The name of the host on which the task is running or was run.

#### DateStarted <DateTime>

The date and time that the task was initiated.

#### Type <Citrix.XDInterServiceTypes.JobType>

The type of task. For newly provisioned VM tasks, this is always "NewVirtualMachine".

#### Metadata <Citrix.MachineCreation.Sdk.Metadata[]>

The list of metadata stored against the task. For new tasks this is empty until metadata is added.

#### WorkflowStatus <System.Workflow.Runtime.WorkflowStatus>

Indicates the status of the workflow that is used to process the task.

#### ProvisioningSchemeName <string>

The name of the provisioning scheme that the task was for.

#### ProvisioningSchemeUid <Guid>

The unique identifier of the provisioning scheme that the task was for.

#### MasterImage <string>

The path (in the hosting unit provider) of the virtual machine snapshot that was used as the master image for the task.

#### IdentityPoolName <string>

The name of the identity pool (from the ADIdentity PowerShell snap-in) that the new provisioning scheme uses.

IdentityPoolUid <guid>

The unique identifier name of the identity pool (from the ADIdentity PowerShell snap-in) that the new provisioning scheme uses.

HostingUnitName <string>

The name of the hosting unit (from the Hosting Unit PowerShell snap-in) that the new provisioning scheme uses.

HostingUnitUid

The unique identifier of the hosting unit (from the Hosting Unit PowerShell snap-in) that the new provisioning scheme uses.

TaskState <Citrix.DesktopUpdateManager.SDK.ProvisionVMState>

The state of the task. This can be any of the following:

Processing

Indicates that the task is in its initial state.

LocatingResources,

Indicates that the task is locating information from other services.

HostingUnitNotFound

Indicates that the required hosting unit could not be located.

ProvisioningSchemeNotFound

Indicates that the required provisioning scheme could not be located.

Provisioning

Indicates that the task is at the provisioning stage.

IdentityPoolNotFound

Indicates that the required identity pool could not be located.

TaskAlreadyRunningForProvisioningScheme

Indicates that the provisioning scheme already has another task running.

HypervisorInMaintenanceMode

Indicates that the hypervisor is not available for normal use.

NoAvailableStorage

Indicates that no storage is available for the hypervisor.

NoAvailableNetwork

Indicates that no network is available for the hypervisor.

## Finalizing

Indicates that the task is finalizing.

## Finished

Indicates that the task is complete.

## FinishedWithErrors

Indicates that the task is complete but there were errors. Specific details of errors are included with each failed virtual machine.

## Removing

Indicates that the task is removing virtual machines from the hypervisor.

## Failed

The job failed for reasons specified in TaskStateInformation.

## Canceled

Indicates that the task was stopped by user intervention (using Stop-ProvTask).

## TaskStateInformation

Provides more detailed information about the current task state.

## VirtualMachinesToCreateCount <int>

The total number of virtual machines that the task is trying to create.

## VirtualMachinesCreatedCount <int>

The number of virtual machines that the task has created so far. Details of the machines that were created are in the CreatedVirtualMachines parameter.

## VirtualMachinesCreationFailedCount <int>

The number of virtual machines that the task has failed to create. Details of the machines that were not created are in the FailedVirtualMachines parameter.

## FailedVirtualMachines <Citrix.DesktopUpdateManager.SDK.VirtualMachineCreation[]>

## ADAccountName <string>

The domain qualified AD Account name of the machine.

## ADAccountSid <string>

The AD account SID of the machine account.

## DiagnosticInformation <Citrix.MachineCreation.Sdk.ExceptionSurrogate[]>

A collection of handled error states which caused the provisioning to fail.

ExceptionType <string>

The type of exception this object represents

Message <string>

The exception message

Details <string>

The full exception content including stack trace

InnerException <Citrix.MachineCreation.Sdk.ExceptionSurrogate>

Information relating to any contributing error state

Status <string>

StatusAdditionalInformation <string>

VMId <string>

The virtual machine identifier within the hypervisor in which the VM resides.

VMName <string>

The display name of the virtual machine within the hypervisor in which the VM resides.

CreatedVirtualMachines <Citrix.DesktopUpdateManager.SDK.VirtualMachineCreation[]>

See FailedVirtualMachines for details of the object parameters.

## Notes

Only one long-running task in each provisioning scheme can be processed at a time.

In the case of failure, the following errors can result.

### Error Codes

-----

#### JobCreationFailed

The requested task could not be started.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation. Communication with the database failed for various reasons.

WorkflowHostUnavailable

The task could not be started because the database connection is inactive.

PermissionDenied

The user does not have administrative rights to perform this operation.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

CommunicationError

An error occurred while communicating with the service.

InvalidParameterCombination

Both PurgeJobOnSuccess and RunAsynchronously were specified.

When running asynchronously, the cmdlet terminates before the job does, so it cannot clean up the completed job.

PermissionDenied

The user does not have administrative rights to perform this operation.

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used, or examine the XenDesktop logs.

The cmdlet is associated with a task of type NewVirtualMachine, and while active will move through the following operations (CurrentOperation field)

ValidatingInputs

CreatingVirtualMachines

ReleasingAccountLocks

Examples

----- **EXAMPLE 1** -----

```
C:\PS>New-provVM -ProvisioningSchemeName MyScheme -ADAccountName "domain\Account"
```

```
TaskId           : 4b49f13a-277c-4cb0-bc40-f088430cfe8a
Active           : False
Host             : DUMTESTDDC
```

```

DateStarted          : 18/05/2010 10:54:32
Type                 : NewVirtualMachine
Metadata             : {}
WorkflowStatus       : Completed
MasterImage          : /Base.vm/Base.snapshot
ProvisioningSchemeName : MyScheme
ProvisioningSchemeUid : 7585f0de-192e-4847-a6d8-22713c3a2f42
CurrentOperation     :
TaskState            : Finished
TaskStateInformation :
HostingUnitUid       : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501
HostingUnitName      : XenHU
IdentityPoolUid      : 03743136-e43b-4a87-af74-ab71686b3c16
IdentityPoolName     : idPool1
VirtualMachinesToCreateCount : 1
VirtualMachinesCreatedCount : 1
VirtualMachinesCreationFailedCount : 0
CreatedVirtualMachines : {DOMAIN\Account$}
FailedVirtualMachines : {}
ProvisioningJob      : 6fa5dc7c-6d49-4616-8682-aeb1580866b3
ProvisioningStatus   : Completed

```

Creates a new virtual machine using the AD account "domain\account" in the provisioning scheme "MyScheme".

----- **EXAMPLE 2** -----

```
C:\PS>New-provVM -ProvisioningSchemeName MyScheme -ADAccountName "domain\Account" -RunAsynchronously
```

Guid

----

```
6dd85fec-96cf-46b1-9cd4-d8ba7d06e85b
```

Creates a new virtual machine using the AD account "domain\account" in the provisioning scheme "MyScheme" asynchronously. To get the task details, use `Get-ProvTask -TaskID <task id>`

i.e.

```
C:\PS>Get-ProvTask -TaskID 6dd85fec-96cf-46b1-9cd4-d8ba7d06e85b
```

```
TaskId : 4b49f13a-277c-4cb0-bc40-f088430cfe8a
```

```
Active : False
```

```
Host : MyHost
```

```
DateStarted : 18/05/2010 10:54:32
```

```
Type : NewVirtualMachine
```

```
Metadata : {}
```

```
WorkflowStatus : Completed
```

```
MasterImage : /Base.vm/Base.snapshot
```

ProvisioningSchemeName : MyScheme  
ProvisioningSchemeUid : 7585f0de-192e-4847-a6d8-22713c3a2f42  
TaskState : Finished  
TaskStateInformation :  
HostingUnitUid : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501  
HostingUnitName : XenHU  
IdentityPoolUid : 03743136-e43b-4a87-af74-ab71686b3c16  
IdentityPoolName : idPool1  
VirtualMachinesToCreateCount : 1  
VirtualMachinesCreatedCount : 1  
VirtualMachinesCreationFailedCount : 0  
CreatedVirtualMachines : {DOMAIN\Account\$}  
FailedVirtualMachines : {}  
ProvisioningJob : 6fa5dc7c-6d49-4616-8682-aeb1580866b3  
ProvisioningStatus : Completed

----- **EXAMPLE 3** -----

```
C:\PS>$accounts = Get-AcctAdAccount -IdentityPool MyPool -State Available  
C:\PS>New-provVM -ProvisioningSchemeName MyScheme -ADAccountName $t -RunAsynchronously  
Creates a new virtual machine using all of the available AD accounts from the identity pool "MyPool" in the provisioning  
scheme "MyScheme" asynchronously. To get the task details, use Get-ProvTask -TaskID <task id>
```

For example:

```
C:\PS>Get-ProvTask -TaskID 6dd85fec-96cf-46b1-9cd4-d8ba7d06e85b  
  
TaskId : 4b49f13a-277c-4cb0-bc40-f088430cfe8a  
  
Active : False  
  
Host : MyHost  
  
DateStarted : 18/05/2010 10:54:32  
  
Type : NewVirtualMachine  
  
Metadata : {}  
  
WorkflowStatus : Completed  
  
MasterImage : /Base.vm/Base.snapshot
```



ProvisioningSchemeName : MyScheme  
ProvisioningSchemeUid : 7585f0de-192e-4847-a6d8-22713c3a2f42  
TaskState : Finished  
TaskStateInformation :  
HostingUnitUid : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501  
HostingUnitName : XenHU  
IdentityPoolUid : 03743136-e43b-4a87-af74-ab71686b3c16  
IdentityPoolName : idPool1  
VirtualMachinesToCreateCount : 1  
VirtualMachinesCreatedCount : 1  
VirtualMachinesCreationFailedCount : 0  
CreatedVirtualMachines : {DOMAIN\Account\$}  
FailedVirtualMachines : {}  
ProvisioningJob : 6fa5dc7c-6d49-4616-8682-aeb1580866b3  
ProvisioningStatus : Completed

# Publish-ProvMasterVmImage

Jan 04, 2017

Update the master image associated with the provisioning scheme.

## Syntax

```
Publish-ProvMasterVmImage [-ProvisioningSchemeName] <String> -MasterImageVM <String> [-DoNotStoreOldImage] [-VhdTemplateSource <String>] [-VhdResultDestination <String>] [-AppScanResultsFile <String>] [-RunAsynchronously] [-PurgeJobOnSuccess] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Publish-ProvMasterVmImage -ProvisioningSchemeUid <Guid> -MasterImageVM <String> [-DoNotStoreOldImage] [-VhdTemplateSource <String>] [-VhdResultDestination <String>] [-AppScanResultsFile <String>] [-RunAsynchronously] [-PurgeJobOnSuccess] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to update the hard disk image used to provision virtual machines. If the provisioning scheme is a 'CleanOnBoot' type, then the next time that virtual machines are started, their hard disks are updated to this new image. Regardless of the 'CleanOnBoot' type, all new virtual machines created after this command will use this new hard disk image.

A background task will be created to remove the old hard disk copies. You can locate and monitor this task using the Get-ProvTask cmdlet.

A snapshot is used rather than a VM, so that the content of the hard disk for the provisioning scheme can be easily determined.

As the snapshot is specified using a path into the Citrix Host Service Powershell Provider, the Citrix Host Service Powershell snap-in must also be loaded for this cmdlet to be used.

The previous hard disk image path is stored into the history (see Get-provSchemeMasterVMImageHistory). The data stored in the history allows for a roll back to be undertaken, to revert to the previous hard disk image if required.

## Related topics

[Get-ProvSchemeMasterVMImageHistory](#)

[Get-ProvScheme](#)

## Parameters

**-ProvisioningSchemeName**<String>

The provisioning scheme to which the hard disk image should be updated.

Required?	true
Default Value	

Accept Pipeline Input?	true (ByPropertyName)
------------------------	-----------------------

**-ProvisioningSchemeUid**<Guid>

The provisioning scheme identifier to which the hard disk image should be updated.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-MasterImageVM**<String>

The path in the hosting unit provider to the virtual machine snapshot that will be used. This identifies the hard disk to be used and the default values for the memory and processors. This must be a path to a Snapshot Item in the same hosting unit that the hosting unit name or hosting unit Uid refers to.

Valid paths are of the format; XDHyp:\HostingUnits\<<HostingUnitName>\<path>\<VmName>.vm\  
<SnapshotName>.snapshot

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-DoNotStoreOldImage**<SwitchParameter>

Specifies that the current image should not be added to the image history list for the provisioning scheme. This is useful when rolling back to a previous image.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-VhdTemplateSource**<String>

A file path to a source VHD template to be used when performing Application Scanning during Image Preparation. The presence of this parameter in conjunction with VhdResultDestination implies that application scanning is to be performed

--	--

Required?	false
Default Value	
Accept Pipeline Input?	false

**-VhdResultDestination**<String>

A file path (including file name) where the VHD disk file containing the results of application scanning should be written.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AppScanResultsFile**<String>

File name to which the results of application scanning should be written.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-RunAsynchronously**<SwitchParameter>

Indicates whether or not the cmdlet should return before it is complete. If specified, the command returns an identifier for the task that was created. You can monitor this task using the get-ProvTask cmdlet.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-PurgeJobOnSuccess**<SwitchParameter>

Indicates that the task history will be removed from the database once the task has finished. This cannot be specified for tasks that are run asynchronously.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

System.Guid

When "RunAsynchronously" is specified, this identifier is returned to provide the task identifier.

System.Management.Automation.PSCustomObject

This object provides details of the task run and contains the following information:

TaskId <Guid>

The identifier for the task that was performed.

Active <Boolean>

Indicates whether the task is still processing or is complete.

Host <string>

The name of the host on which the task is running or was run.

DateStarted <DateTime>

The date and time that the task was initiated.

Type <Citrix.XDInterServiceTypes.JobType>

The type of the task. For new provisioning scheme tasks this is always "NewProvisioningScheme".

Metadata <Citrix.MachineCreation.Sdk.Metadata[]>

The list of metadata stored against the task. For new tasks this will be empty until metadata is added.

WorkflowStatus <System.Workflow.Runtime.WorkflowStatus>

Indicates the status of the workflow that is being used to process the task.

ProvisioningSchemeName <string>

The name of the provisioning scheme that the task was for.

ProvisioningSchemeUid <Guid>

The unique identifier of the provisioning scheme that the task was for.

MasterImage <string>

The path (in the hosting unit provider) of the virtual machine snapshot that was used as the master image for the task.

IdentityPoolName <string>

The name of the identity pool (from the ADIdentity PowerShell snap-in) that the new provisioning scheme will use.

IdentityPoolUid <guid>

The unique identifier name of the identity pool (from the ADIdentity PowerShell snap-in) that the new provisioning scheme will use.

HostingUnitName <string>

The name of the hosting unit (from the Hosting Unit PowerShell snap-in) that the new provisioning scheme will use.

HostingUnitUid

The unique identifier of the hosting unit (from the Hosting Unit PowerShell snap-in) that the new provisioning scheme will use.

TaskState <Citrix.DesktopUpdateManager.SDK.NewProvisioningSchemeState>

The state of the task. This can be any of the following:

Processing

Indicates that the task has just begun and has not done anything yet.

LocatingResources,

Indicates that the workflow is locating resources from other services.

HostingUnitNotFound

Indicates that the task failed because the required hosting unit could not be located.

VirtualMachineSnapshotNotFound

Indicates that the task failed because the required snapshot could not be located.

ConsolidatingMasterImage

Indicates that the task is consolidating the master image.

ReplicatingConsolidatedImageToAllStorage

Indicates that the task is replicating the consolidated master image.

StoringProvisioningScheme

Indicates that the task is storing the provisioning scheme data to the database.

Finished

Indicates that the task has completed with no errors.

ProvisioningSchemeAlreadyExists

Indicates that the task failed because a provisioning scheme with the same name already exists.

IdentityPoolNotFound

Indicates that the task failed because the identity pool specified could not be found.

MasterVMImagesNotPartOfProvisioningSchemeHostingUnit,

Indicates that the hosting unit that the master image originated from is not the same hosting unit that the provisioning scheme is defined to use.

MasterVmImagesNotASnapshot

Indicates that the task failed because the master VM Image path does not refer to a Snapshot item.

ProvisioningSchemeNotFound

Could not find a provisioning scheme with the specified name.

TaskAlreadyRunningForProvisioningScheme

A task for a provisioning scheme with the same name is already running.

InvalidMasterVMConfiguration

The VM snapshot specified as the master had an invalid configuration.

InvalidMasterVMState

The VM snapshot specified as the master is currently in an invalid state.

InsufficientResources

Indicates that the task failed because the hypervisor did not have enough resources to complete the task.

StorageNotFound

Indicates that no associated storage was found in the hosting unit.

Canceled

Indicates that the task was stopped by user intervention (using Stop-ProvTask)

TaskStateInformation <string>

Holds string data providing more details about the current task state.

TaskProgress

The progress of the task 0-100%.

Notes

Only one long running task per provisioning scheme can be processed at any one time.

In the case of failure, the following errors can result.

Error Codes

-----

JobCreationFailed

The requested task could not be started.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

WorkflowHostUnavailable



The task could not be started because the database connection is inactive.

CommunicationError

An error occurred while communicating with the service.

PermissionDenied

The user does not have administrative rights to perform this operation.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

The cmdlet is associated with a task of type PublishImage, and while active will move through the following operations (CurrentOperation field)

ValidatingInputs

ConsolidatingMasterImage

PreparingMasterImage

ReplicatingMasterImage

CommittingScheme

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Publish-ProvMasterVMImage -ProvisioningSchemeName MyScheme -MasterImageVM XDHyp:\H  
ostingUnits\HostUnit1\RhoneCC_baseXP.vm\base.snapshot
```

```
TaskId          : 248f102f-073f-45fe-aea9-1819a6d6dd1f  
Active          : False  
Host            : MyHost  
DateStarted     : 17/05/2010 17:37:57  
Type            : PublishImage  
Metadata        : {}  
WorkflowStatus  : Completed  
ProvisioningSchemeUid : 7585f0de-192e-4847-a6d8-22713c3a2f42  
ProvisioningSchemeName : MyScheme  
MasterImage     : /RhoneCC_baseXP.vm/base.snapshot  
HostingUnitName : HostUnit1  
HostingUnitUid  : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501  
ADIdentityPoolName :  
ADIdentityPoolUid : 03743136-e43b-4a87-af74-ab71686b3c16
```

CurrentOperation :

TaskState : Finished

TaskStateInformation :

Updates the master hard disk image for the provisioning Scheme "MyScheme" to use the "base.snapshot" hard disk image.

# Remove-ProvScheme

Jan 04, 2017

Removes a provisioning scheme

## Syntax

```
Remove-ProvScheme [-ProvisioningSchemeName] <String> [-ForgetVM] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ProvScheme -ProvisioningSchemeUid <Guid> [-ForgetVM] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability to remove a provisioning Scheme. The provisioning scheme must not contain any VMs unless the 'ForgetVM' option is specified.

If 'ForgetVM' is not specified, a cmdlet task is created that runs in the background to remove the hard disk copies that have been created for the provisioning scheme in hypervisor storage. Use the Get-ProvTask command to monitor the progress of this task.

## Related topics

[Get-ProvTask](#)

[New-ProvScheme](#)

## Parameters

**-ProvisioningSchemeName**<String>

The name of the provisioning scheme to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ProvisioningSchemeUid**<Guid>

The unique identifier for the provisioning scheme to be removed.

Required?	true
Default Value	

Accept Pipeline Input?	true (ByPropertyName)
------------------------	-----------------------

**-ForgetVM<SwitchParameter>**

Indicates whether or not the VMs in the provisioning scheme should be left in the hypervisor and only the data held in the Machine Creation Services removed. If this is specified, it is up to the administrator of the hypervisor to remove the VMs and hard disk images using the tools provided by the hypervisor itself.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-LoggingId<Guid>**

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

Citrix.MachineCreation.Sdk.ProvisioningScheme

This object provides details of the provisioning scheme and contains the following information:

#### ProvisioningSchemeUid

The unique identifier for the provisioning scheme.

#### ProvisioningSchemeName

The name of the provisioning scheme.

#### CpuCount

The number of processors that VMs will be created with when using this scheme.

#### MemoryMB

The maximum amount of memory that VMs will be created with when using this scheme.

#### MasterImageVM

The path within the hosting unit provider to the VM or snapshot of which the scheme is currently using a copy.

#### MasterImageVMDate

The date and time that the copy of the VM image was made for the scheme.

#### IdentityPoolUid

The unique identifier of the identity pool (from the ADIdentity PowerShell snap-in) that the scheme uses.

#### IdentityPoolName

The name of the identity pool (from the ADIdentity PowerShell snap-in) that the scheme uses.

#### HostingUnitUid

The unique identifier of the hosting unit (from the Hosting Unit PowerShell snap-in) that the new provisioning scheme will use.

#### HostingUnitName

The name of the hosting unit (from the hosting unit PowerShell snap-in) that the new provisioning scheme will use.

#### CleanOnBoot

Indicates whether the VMs created are to be reset to a clean state on each boot.

#### TaskId

The identifier of any current task that is running for the provisioning scheme.

#### Notes

If the hosting unit referenced by the provisioning scheme no longer exists (i.e. it has been removed using the Hosting Unit PowerShell snap-in), the provisioning scheme data is deleted from the database without errors. However, the hard disks associated with the provisioning scheme cannot be removed and remain in the hypervisor.

In the case of failure, the following errors can result.

## Error Codes

-----

### IllegalParameter

One or more parameters are illegal or are not specified.

### ProvisioningSchemeNotFound

The specified provisioning scheme could not be located.

### UnableToRemoveProvisioningSchemeDueToAssociatedVM

The provisioning scheme contained VMs and the 'ForgetVM' parameter was not specified.

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

### CommunicationError

An error occurred while communicating with the service.

### PermissionDenied

The user does not have administrative rights to perform this operation.

### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

The cmdlet is associated with a task of type DisusedImageCleanUp, and while active will move through the following operations (CurrentOperation field)

Running

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Remove-ProvScheme -ProvisioningSchemeName $provScheme.ProvisioningSchemeName
Remove the empty provisioning scheme by name.
```

# Remove-ProvSchemeControllerAddress

Jan 04, 2017

Removes metadata from a provisioning scheme.

## Syntax

```
Remove-ProvSchemeControllerAddress [-ProvisioningSchemeName] <String> [-ControllerAddress] <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [-CommonParameters]
```

```
Remove-ProvSchemeControllerAddress -ProvisioningSchemeUid <Guid> -ControllerAddress <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [-CommonParameters]
```

## Detailed Description

Removes the specified controller addresses from the specified object. Attempting to remove an address not present writes an error record to the pipeline.

## Related topics

[Get-ProvScheme](#)

[Add-ProvSchemeControllerAddress](#)

## Parameters

**-ProvisioningSchemeName**<String>

The name of the provisioning scheme from which metadata is to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ProvisioningSchemeUid**<Guid>

The identifier of the provisioning scheme from which metadata is to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-ControllerAddress**<String[]>

Specifies the array of DNS names to be removed from the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	



Accept Pipeline Input?	false
------------------------	-------

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Input Type**

Citrix.MachineCreation.Sdk.ProvisioningScheme You can pipe an object containing a parameter called 'ProvisioningSchemeName' to Remove-ProvSchemeMetadata.

**Notes**

In the case of failure, the following errors can result.

**Error Codes**

-----

**ProvisioningSchemeNotFound**

The specified provisioning scheme could not be located.

**ControllerAddressNotFound**

The specified address was not associated with the provisioning scheme.

**DatabaseError**

An error occurred in the service while attempting a database operation.

**DatabaseNotConfigured**

The operation could not be completed because the database for the service is not configured.

**ServiceStatusInvalidDb**

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

**CommunicationError**

An error occurred while communicating with the service.

**PermissionDenied**

The user does not have administrative rights to perform this operation.

**ConfigurationLoggingError**

The operation could not be performed because of a configuration logging error

**ExceptionThrown**

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

**Examples**

----- **EXAMPLE 1** -----

```
C:\PS>Get-ProvScheme -ProvisioningSchemeName scheme1 | Remove-ProvSchemeControllerAddress
```

```
ProvisioningSchemeUid : 7585f0de-192e-4847-a6d8-22713c3a2f42
ProvisioningSchemeName : Scheme1
CpuCount : 1
```

```
MemoryMB      : 1024
MasterImageVM : Base.vm/Base.snapshot
MasterImageVMDate : 17/05/2010 09:53:40
IdentityPoolUid : 03743136-e43b-4a87-af74-ab71686b3c16
IdentityPoolName : idPool1
HostingUnitUid : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501
HostingUnitName : HostUnit1
CleanOnBoot    : True
TaskId         : 00000000-0000-0000-0000-000000000000
Metadata       : {}
ControllerAddress : {}
Remove all controller addresses from the provisioning scheme with the name "scheme1".
```

----- **EXAMPLE 2** -----

```
C:\PS>Remove-ProvSchemeControllerAddress -ProvisioningSchemeUid "01a4a008-8ce8-4165-ba9c-cdf15a6b0501" -ControllerAddress (ddcA.citrix.com,ddcC.citrix2.com)
```

```
ProvisioningSchemeUid : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501
ProvisioningSchemeName : Scheme2
CpuCount              : 1
MemoryMB              : 1024
MasterImageVM         : Base.vm/Base.snapshot
MasterImageVMDate     : 17/05/2010 09:53:40
IdentityPoolUid       : 03743136-e43b-4a87-af74-ab71686b3c16
IdentityPoolName      : idPool1
HostingUnitUid        : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501
HostingUnitName       : HostUnit1
CleanOnBoot           : True
TaskId                : 00000000-0000-0000-0000-000000000000
Metadata              : {}
ControllerAddress      : {ddcB.citrix.com}
```

Remove a subset of the controller address list from the provisioning scheme with the identifier "01a4a008-8ce8-4165-ba9c-cdf15a6b0501".

# Remove-ProvSchemeMasterVMImageHistory

Jan 04, 2017

Removes the history of provisioning scheme master image VMs.

## Syntax

```
Remove-ProvSchemeMasterVMImageHistory [-ProvisioningSchemeName] <String> [-VMImageHistoryUid <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeMasterVMImageHistory -ProvisioningSchemeUid <Guid> [-VMImageHistoryUid <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeMasterVMImageHistory -VMImageHistoryUid <Guid> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove the record of previously used master VMs or snapshots for provisioning schemes.

## Related topics

[Get-ProvSchemeMasterVMImageHistory](#)

[Publish-ProvMasterVMImage](#)

## Parameters

**-ProvisioningSchemeName**<String>

The name of the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ProvisioningSchemeUid**<Guid>

The unique identifier of the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-VMImageHistoryUid**<Guid>

The unique identifier for the Image History item.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Notes

In the case of failure, the following errors can result.

#### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
c:\PS>Remove-ProvSchemeMasterVMImageHistory -ProvisioningSchemeName MyScheme  
Removes all history for the provisioning scheme called "MyScheme".
```

##### ----- EXAMPLE 2 -----

```
c:\PS>Get-ProvScheme | Remove-ProvSchemeMasterVMImageHistory  
Removes all history for all provisioning schemes.
```

# Remove-ProvSchemeMetadata

Jan 04, 2017

Removes metadata from the given ProvisioningScheme.

## Syntax

```
Remove-ProvSchemeMetadata [-ProvisioningSchemeUid] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeMetadata [-ProvisioningSchemeUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeMetadata [-ProvisioningSchemeName] <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeMetadata [-ProvisioningSchemeName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeMetadata [-InputObject] <ProvisioningScheme[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeMetadata [-InputObject] <ProvisioningScheme[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given ProvisioningScheme.

## Related topics

[Add-ProvSchemeMetadata](#)

[Set-ProvSchemeMetadata](#)

## Parameters

**-ProvisioningSchemeUid**<Guid>

Id of the ProvisioningScheme

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-ProvisioningSchemeName**<String>

Name of the ProvisioningScheme

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<ProvisioningScheme[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create

high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.



PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ProvScheme | Remove-ProvSchemeMetadata  
Remove all metadata from all ProvisioningScheme objects.
```

# Remove-ProvSchemeScope

Jan 04, 2017

Remove the specified ProvisioningScheme(s) from the given scope(s).

## Syntax

```
Remove-ProvSchemeScope [-Scope] <String[]> -InputObject <ProvisioningScheme[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeScope [-Scope] <String[]> -ProvisioningSchemeUid <Guid[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeScope [-Scope] <String[]> -ProvisioningSchemeName <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The RemoveProvSchemeScope cmdlet is used to remove one or more ProvisioningScheme objects from the given scope(s).

There are multiple parameter sets for this cmdlet, allowing you to identify the ProvisioningScheme objects in different ways:

- ProvisioningScheme objects can be piped in or specified by the InputObject parameter
- The ProvisioningSchemeUid parameter specifies objects by ProvisioningSchemeUid
- The ProvisioningSchemeName parameter specifies objects by ProvisioningSchemeName (supports wildcards)

To remove a ProvisioningScheme from a scope you need permission to change the scopes of the ProvisioningScheme.

If the ProvisioningScheme is not in a specified scope, that scope will be silently ignored.

## Related topics

[Add-ProvSchemeScope](#)

[Get-ProvScopedObject](#)

## Parameters

**-Scope**<String[]>

Specifies the scopes to remove the objects from.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-InputObject**<ProvisioningScheme[]>

Specifies the ProvisioningScheme objects to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-ProvisioningSchemeUid**<Guid[]>

Specifies the ProvisioningScheme objects to be removed by ProvisioningSchemeUid.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-ProvisioningSchemeName**<String[]>

Specifies the ProvisioningScheme objects to be removed by ProvisioningSchemeName.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or

an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

None

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### ScopeNotFound

One of the specified scopes was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command with the specified objects or scopes.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Remove-ProvSchemeScope Finance -ProvisioningSchemeUid 6702C5D0-C073-4080-A0EE-EC74CB537C52
```

Removes a single ProvisioningScheme from the 'Finance' scope.

----- **EXAMPLE 2** -----

```
c:\PS>Remove-ProvSchemeScope Finance,Marketing -ProvisioningSchemeUid 6702C5D0-C073-4080-A0EE-EC74CB537C52
```

Removes a single ProvisioningScheme from multiple scopes.

----- **EXAMPLE 3** -----

```
c:\PS>Get-ProvScheme | Remove-ProvSchemeScope Finance
```

Removes all visible ProvisioningScheme objects from the 'Finance' scope.

----- **EXAMPLE 4** -----

```
c:\PS>Remove-ProvSchemeScope Finance -ProvisioningSchemeName A*
```

Removes ProvisioningScheme objects with a name starting with an 'A' from the 'Finance' scope.

# Remove-ProvServiceConfigurationData

Jan 04, 2017

Removes configuration data from the service.

## Syntax

```
Remove-ProvServiceConfigurationData [[-Name] <String>] [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability to remove data from the MachineCreation Service configuration data.

## Related topics

[Set-ProvServiceConfigurationData](#)

[Get-ProvServiceConfigurationData](#)

## Parameters

**-Name**<String>

The name of the configuration data item to remove.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name

or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Notes

In the case of failure the following errors can be produced.

### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

An error occurred while communicating with the service.

ExceptionThrown An unexpected error occurred. To locate more details see the Windows event logs on the controller being used, or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS>Remove-ProvServiceConfigurationData -Name "MyName"
```

Removes the configuration data item with name "MyName".



# Remove-ProvServiceMetadata

Jan 04, 2017

Removes metadata from the given Service.

## Syntax

```
Remove-ProvServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ProvServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ProvServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ProvServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Service.

## Related topics

[Set-ProvServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<Service[]>

Objects to which metadata is to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
-----------	-------

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ProvService | % { Remove-ProvServiceMetadata -Map $_.MetadataMap }  
Remove all metadata from all Service objects.
```

# Remove-ProvTask

Jan 04, 2017

Removes from the database completed tasks for the MachineCreation Service.

## Syntax

```
Remove-ProvTask [-TaskId] <Guid> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables completed tasks that have run within the MachineCreation Service to be removed from the database.

## Related topics

[Get-ProvTask](#)

[Add-ProvTaskMetadata](#)

[Remove-ProvTaskMetadata](#)

## Parameters

**-TaskId**<Guid>

Specifies the identifier for the task to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Input Type

System.Management.Automation.PSObject Objects containing the TaskId parameter can be piped to the Remove-ProvTask command.

### Notes

If the command fails, the following errors can result.

#### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### InvalidWorkflow

The specified task could not be found.

#### InvalidWorkflowState

The task was not in an appropriate state for the requested operation.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ProvTask -active $false | Remove-ProvTask
```

Success

Remove entries for all completed tasks from the MachineCreation Service.

# Remove-ProvTaskMetadata

Jan 04, 2017

Removes metadata from the given Task.

## Syntax

```
Remove-ProvTaskMetadata [-TaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ProvTaskMetadata [-TaskId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ProvTaskMetadata [-InputObject] <Task[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ProvTaskMetadata [-InputObject] <Task[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Task.

## Related topics

[Add-ProvTaskMetadata](#)

[Set-ProvTaskMetadata](#)

[Get-ProvTask](#)

[Remove-ProvTask](#)

[Stop-ProvTask](#)

[Switch-ProvTask](#)

## Parameters

**-TaskId**<Guid>

Id of the Task

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<Task[]>



Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-ProvTask | Remove-ProvTaskMetadata
```

Remove all metadata from all Task objects.

# Remove-ProvVM

Jan 04, 2017

Removes virtual machines.

## Syntax

```
Remove-ProvVM [-ProvisioningSchemeName] <String> -VMName <String[]> [-ForgetVM] [-RunAsynchronously] [-PurgeJobOnSuccess] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvVM [-ProvisioningSchemeUid] <Guid> -VMName <String[]> [-ForgetVM] [-RunAsynchronously] [-PurgeJobOnSuccess] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Lets you remove VMs from the Machine Creation Services and the hypervisor that they run on.

## Related topics

[Get-ProvVM](#)

[New-ProvVM](#)

## Parameters

**-ProvisioningSchemeName**<String>

The name of the provisioning scheme from which virtual machines will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ProvisioningSchemeUid**<Guid>

The unique identifier for the provisioning scheme from which the virtual machines are removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-VMName**<String[]>

List of VM names that will be removed from the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-ForgetVM**<SwitchParameter>

The named VMs will only be removed from the provisioning scheme database, but will remain in the hypervisor.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-RunAsynchronously**<SwitchParameter>

Indicates whether or not the command returns before it is complete. If this is specified, the command returns an identifier for the task that was created. This task can be monitored using the get-ProvTask command.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-PurgeJobOnSuccess**<SwitchParameter>

Indicates that the task history is removed from the database when the task finishes. This cannot be specified for tasks that are run asynchronously.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. When a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Guid

When the RunAsynchronously identifier is specified, this GUID is returned and provides the task identifier.

System.Management.Automation.PSCustomObject

This object provides details of the task that was run and contains the following information:

TaskId <Guid>

The identifier for the task that was performed.

Active <Boolean>

Indicates whether the task is still processing or is complete.

Host <string>

The name of the host on which the task is running or was run.

DateStarted <DateTime>

The date and time that the task was initiated.

Type <Citrix.XDInterServiceTypes.JobType>

The type of task. For new remove-VM tasks, this is always "RemoveVirtualMachine".

Metadata <Citrix.MachineCreation.Sdk.Metadata[]>

The list of metadata stored for the task. For new tasks, this is empty until metadata is added.

WorkflowStatus <System.Workflow.Runtime.WorkflowStatus>

Indicates the status of the workflow that is used to process the task.

ProvisioningSchemeName <string>

The name of the provisioning scheme that the task is for.

ProvisioningSchemeUid <Guid>

The unique identifier of the provisioning scheme that the task is for.

TaskState <Citrix.DesktopUpdateManager.SDK.ProvisionVMState>

The state of the task. This can be any of the following:

Processing

Indicates that the task is in its initial state.

LocatingResources,

Indicates that the task is locating information from other services.

HostingUnitNotFound

Indicates that the required hosting unit could not be located.

ProvisioningSchemeNotFound

Indicates that the required provisioning scheme could not be located.

Provisioning

Indicates that the task is at the provisioning stage.

IdentityPoolNotFound

Indicates that the required identity pool could not be located.

TaskAlreadyRunningForProvisioningScheme

Indicates that the provisioning scheme already has another task running.

Finalizing

Indicates that the task is finalizing.

Finished

Indicates that the task is complete.

FinishedWithErrors

Indicates that the task is complete but there were errors. Specific details of errors are included with each failed virtual machine.

Removing

Indicates that the task is removing virtual machines from the hypervisor.

Failed

Job failed for reasons specified in TaskStateInformation.

Canceled

Indicates that the task was stopped by user intervention (using Stop-ProvTask)

TaskStateInformation

Provides more detailed information about the current task state.

VirtualMachinesToRemoveCount <int>

The total number of virtual machines that the task is trying to remove.

VirtualMachinesRemovedCount <int>

The number of virtual machines that the task has removed so far. Details of the machines that have been removed are in the RemovedVirtualMachines parameter.

VirtualMachinesFailedCount <int>

The number of virtual machines that the task has failed to remove. Details of the machines that failed are in the FailedVirtualMachines parameter.

FailedVirtualMachines <Citrix.DesktopUpdateManager.SDK.VirtualMachineCreation[]>

ADAccountName <string>

The domain-qualified AD Account name of the machine.

ADAccountSid <string>

The AD account SID of the machine account.

DiagnosticInformation <Citrix.MachineCreation.Sdk.ExceptionSurrogate[]>

A collection of handled error states which caused the provisioning to fail.

ExceptionType <string>

The type of exception this object represents

Message <string>



The exception message

Details <string>

The full exception content including stack trace

InnerException <Citrix.MachineCreation.Sdk.ExceptionSurrogate>

Information relating to any contributing error state

Status <string>

StatusAdditionalInformation <string>

VMId <string>

The virtual machine identifier in the hypervisor.

VMName <string>

The display name of the virtual machine in the hypervisor.

RemovedVirtualMachines <Citrix.DesktopUpdateManager.SDK.VirtualMachineCreation[]>

See FailedVirtualMachines for details of the object parameters.

ProgressEstimator

Gives an estimate of the number of virtual machines processed, averaging over virtual machines that were both partly and completely processed.

Notes

Only one long-running task for each provisioning scheme can be processed at a time.

This task still operates if the hosting unit or VMs in the hypervisor are missing. This removes the data from the Citrix Machine Creation Services, but the VMs remain in the hypervisor.

In the case of failure, the following errors can result.

Error Codes

-----

JobCreationFailed

The requested task could not be started.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation. Communication with the database failed for various reasons.

#### WorkflowHostUnavailable

The task could not be started because the database connection is inactive.

#### CommunicationError

An error occurred while communicating with the service.

#### InvalidParameterCombination

Both `PurgeJobOnSuccess` and `RunAsynchronously` were specified. When running asynchronously, the cmdlet terminates before the job does, so it cannot clean up the completed job.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

The cmdlet is associated with a task of type `RemoveVirtualMachine`, and while active will move through the following operations (`CurrentOperation` field)

#### ValidatingInputs

#### RemovingVirtualMachines

### Examples

#### ----- EXAMPLE 1 -----

```
c:\PS>,(Get-ProvVM -ProvisioningSchemeName XenPS) | Remove-ProvVM XenPS
```

```
TaskId           : cfb506a5-cc7e-4a49-ac7b-dd960029d0d3
Active           : False
Host             : DDC
DateStarted      : 10/10/2012 16:39:45
Metadata         : {}
WorkflowStatus   : Completed
ProvisioningSchemeUid : e1afc8fb-3f52-42ea-9a17-305fdb0b6ee4
ProvisioningSchemeName : XenPS
TaskState        : Finished
```

TaskStateInformation :  
VirtualMachinesToRemoveCount : 2  
RemovedVirtualMachines : {XD\IP0001\$, XD\IP0002\$}  
FailedVirtualMachines : {}  
VirtualMachinesRemovedCount : 2  
VirtualMachinesFailedCount : 0  
ProgressEstimator : 2  
Type : RemoveVirtualMachine  
Status : Finished  
CurrentOperation :  
TaskProgress : 100  
TaskExpectedCompletion : 10/10/2012 16:39:50  
LastUpdateTime : 10/10/2012 16:39:50  
ActiveElapsedTime : 5  
DateFinished : 10/10/2012 16:39:50  
TerminatingError :  
Remove all VMs from provisioning scheme XenPS

# Rename-ProvScheme

Jan 04, 2017

Renames a provisioning scheme.

## Syntax

```
Rename-ProvScheme [-ProvisioningSchemeName] <String> [-NewProvisioningSchemeName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Rename-ProvScheme -ProvisioningSchemeUid <Guid> [-NewProvisioningSchemeName] <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to change the name of an existing provisioning scheme. The unique identifier for the provisioning scheme never changes after it has been created so, regardless of any name changes, the provisioning scheme can always be identified by its unique identifier.

## Related topics

[New-ProvScheme](#)

[Set-ProvScheme](#)

[Get-ProvScheme](#)

[Test-ProvSchemeNameAvailable](#)

## Parameters

**-ProvisioningSchemeName**<String>

The current name of the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-ProvisioningSchemeUid**<Guid>

The identifier for the provisioning scheme that is to be renamed.

Required?	true
Default Value	

Accept Pipeline Input?	false
------------------------	-------

**-NewProvisioningSchemeName**<String>

The new name for the provisioning scheme. This must be a name that is not being used by an existing provisioning scheme, and it must not contain any of the following characters \/:#.\*?=<>|[]()''''

Required?	true
Default Value	
Accept Pipeline Input?	false

**-PassThru**<SwitchParameter>

Defines whether or not the command returns a result showing the new state of the updated identity pool.

Required?	false
Default Value	true
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.

Accept Pipeline Input?	false
------------------------	-------

## Return Values

Citrix.MachineCreation.Sdk.ProvisioningScheme

This object provides details of the provisioning scheme and contains the following information:

ProvisioningSchemeUid

The unique identifier for the provisioning scheme.

ProvisioningSchemeName

The name of the provisioning scheme.

CpuCount

The number of processors that VMs will be created with when using this scheme.

MemoryMB

The maximum amount of memory that VMs will be created with when using this scheme.

MasterImageVM

The path within the hosting unit provider to the VM or snapshot of which the scheme is currently using a copy.

MasterImageVMDate

The date and time that the copy of the VM image was made for the scheme.

IdentityPoolUid

The unique identifier of the identity pool (from the ADIdentity PowerShell snap-in) that the scheme uses.

IdentityPoolName

The name of the identity pool (from the ADIdentity PowerShell snap-in) that the scheme uses.

HostingUnitUid

The unique identifier of the hosting unit (from the Hosting Unit PowerShell snap-in) that the new provisioning scheme will use.

HostingUnitName

The name of the hosting unit (from the Hosting Unit PowerShell snap-in) that the new provisioning scheme will use.

CleanOnBoot

Indicates whether the VMs created are to be reset to a clean state on each boot.

TaskId

The identifier of any current task that is running for the provisioning scheme.

## Metadata

The metadata for the provisioning scheme.

## Notes

In the case of failure, the following errors can result.

## Error Codes

-----

### ProvisioningSchemeNotFound

The specified provisioning scheme could not be located.

### ProvisioningSchemeNameAlreadyExists

A provisioning scheme with the same name as the new provisioning scheme name already exists.

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

### CommunicationError

An error occurred while communicating with the service.

### PermissionDenied

The user does not have administrative rights to perform this operation.

### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

### ExceptionThrown

An unexpected error occurred. To locate more details, see the windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS>Rename-ProvScheme -provisioningSchemeName CurrentName -NewProvisioningSchemeName NewName  
Renames a provisioning scheme from "currentName" to "NewName".
```

----- **EXAMPLE 2** -----

```
C:\PS>Rename-ProvScheme -provisioningSchemeName CurrentName -NewProvisioningSchemeName NewName -PassThru
```

```
ProvisioningSchemeUid : 7585f0de-192e-4847-a6d8-22713c3a2f42
ProvisioningSchemeName : NewName
CpuCount              : 1
MemoryMB              : 1024
MasterImageVM         : /Base.vm
MasterImageVMDate     : 17/05/2010 08:27:50
IdentityPoolUid       : 03743136-e43b-4a87-af74-ab71686b3c16
IdentityPoolName      : IdPool1
HostingUnitUid        : 01a4a008-8ce8-4165-ba9c-cdf15a6b0501
HostingUnitName       : XenHU
CleanOnBoot           : True
TaskId                :
Metadata              : {}
```

Renames a provisioning scheme from "currentName" to "NewName" and displays the resulting state.



# Reset-ProvServiceGroupMembership

Jan 04, 2017

Reloads the access permissions and configuration service locations for the MachineCreation Service.

## Syntax

```
Reset-ProvServiceGroupMembership [-ConfigServiceInstance] <ServiceInstance[]> [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

## Detailed Description

Enables you to reload MachineCreation Service access permissions and configuration service locations. The Reset-ProvServiceGroupMembership command must be run on at least one instance of the service type (Prov) after installation and registration with the configuration service. Without this operation, the MachineCreation services will be unable to communicate with other services in the XenDesktop deployment. When the command is run, the services are updated when additional services are added to the deployment, provided that the configuration service is not stopped. The Reset-ProvServiceGroupMembership command can be run again to refresh this information if automatic updates do not occur when new services are added to the deployment. If more than one configuration service instance is passed to the command, the first instance that meets the expected service type requirements is used.

## Related topics

### Parameters

**-ConfigServiceInstance**<ServiceInstance[]>

Specifies the configuration service instance object that represents the service instance for the type 'InterService' that references a configuration service for the deployment.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	
Accept Pipeline Input?	

Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Input Type

Citrix.MachineCreation.Sdk.ServiceInstance[] Service instances containing a ServiceInstance object that refers to the central configuration service interservice interface can be piped to the Reset-ProvServiceGroupMembership command.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoSuitableServiceInstance

None of the supplied service instance objects were suitable for resetting service group membership.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-ProvServiceGroupMembership
```

Reset the service group membership for a service in a deployment where the configuration service is configured and running on the same machine as the service.

----- **EXAMPLE 2** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config -AdminAddress OtherServer.example.com | Reset-ProvServiceGroupmembership
```

Reset the service group membership for a service in a deployment where the configuration service that is configured and running on a machine named 'OtherServer.example.com'.

# Set-ProvDBConnection

Jan 04, 2017

Configures a database connection for the MachineCreation Service.

## Syntax

```
Set-ProvDBConnection [-DBConnection] <String> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Configures a connection to a database in which the MachineCreation Service can store its state. The service will attempt to connect and start using the database immediately after the connection is configured. The database connection string is updated to the specified value regardless of whether it is valid or not. Specifying an invalid connection string prevents a service from functioning until the error is corrected.

After a connection is configured, you cannot alter it without first clearing it (by setting the connection to \$null).

You do not need to configure a database connection to use this command.

## Related topics

[Get-ProvServiceStatus](#)

[Get-ProvDBConnection](#)

[Test-ProvDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be used by the MachineCreation Service. Passing in \$null will clear any existing database connection configured.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Force**<SwitchParameter>

If present, allows the local administrator to set the connection string to null when there are problems contacting the database or other services.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and

Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

#### Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Set-ProvDBConnection command returns an object containing the status of the MachineCreation Service together with extra diagnostics information.

DBUnconfigured

The MachineCreation Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the MachineCreation Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the MachineCreation Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the MachineCreation Service currently in use is incompatible with the version of the MachineCreation Service schema on the database. Upgrade the MachineCreation Service to a more recent version.

DBOlderVersionThanService

The version of the MachineCreation Service schema on the database is incompatible with the version of the MachineCreation Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The MachineCreation Service is running and is connected to a database containing a valid schema.

Failed

The MachineCreation Service has failed.

Unknown

The status of the MachineCreation Service cannot be determined.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidDBConnectionString

The database connection string has an invalid format.

DatabaseConnectionDetailsAlreadyConfigured

There was already a database connection configured. After a configuration is set, it can only be set to \$null.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-ProvDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Configures a database connection string for the MachineCreation Service.

----- **EXAMPLE 2** -----

```
c:\PS>Set-ProvDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Configures an invalid database connection string for the MachineCreation Service.

# Set-ProvScheme

Jan 04, 2017

Changes the parameter values for a provisioning scheme.

## Syntax

```
Set-ProvScheme [-ProvisioningSchemeName] <String> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-ServiceOffering <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme [-ProvisioningSchemeName] <String> -IdentityPoolUid <Guid> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-ServiceOffering <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme [-ProvisioningSchemeName] <String> -IdentityPoolName <String> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-ServiceOffering <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme -ProvisioningSchemeUid <Guid> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-ServiceOffering <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme -ProvisioningSchemeUid <Guid> -IdentityPoolName <String> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-ServiceOffering <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme -ProvisioningSchemeUid <Guid> -IdentityPoolUid <Guid> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-ServiceOffering <String>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to update the parameters of an existing provisioning scheme.

This allows the following parameters to be updated:

Number of CPUs that will be used for VMs created from the provisioning scheme. Maximum amount of memory that will be used for VMs created from the provisioning scheme.

To change the name of the provisioning scheme see [Rename-ProvScheme](#).

## Related topics

[New-ProvScheme](#)

[Remove-ProvScheme](#)

[Get-ProvScheme](#)

[Rename-ProvScheme](#)

## Parameters

**-ProvisioningSchemeName**<String>

The name of the provisioning scheme to be updated.

Required?	true
-----------	------

Default Value	
Accept Pipeline Input?	false

**-IdentityPoolUid**<Guid>

The identifier of an identity pool to associate with the provisioning scheme, replacing the present one.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-IdentityPoolName**<String>

The name of an identity pool to associate with the provisioning scheme, replacing the present one.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-ProvisioningSchemeUid**<Guid>

The identifier of the provisioning scheme to be updated.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-VMCpuCount**<Int32>

The number of processors that virtual machines created from the provisioning scheme should use.

Required?	false
-----------	-------



Default Value	
Accept Pipeline Input?	false

**-VMMemoryMB**<Int32>

The maximum amount of memory that virtual machines created from the provisioning scheme should use.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ServiceOffering**<String>

The name of the cloud service offering to use when creating machines.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-PassThru**<SwitchParameter>

Returns the affected record. By default, this cmdlet does not generate any output.

Required?	false
Default Value	False
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
-----------	-------

Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

Citrix.MachineCreation.Sdk.ProvisioningScheme

This object provides details of the provisioning scheme and contains the following information:

ProvisioningSchemeUid

ProvisioningSchemeName

The name of the provisioning scheme.

CpuCount

The number of processors that VMs will be created with when using this scheme.

MemoryMB

The maximum amount of memory that VMs will be created with when using this scheme.

MasterImageVM

The path within the hosting unit provider to the VM or snapshot of which the scheme is currently using a copy.

MasterImageVMDate

The date and time that the copy of the VM image was made for the scheme.

IdentityPoolUid

The unique identifier of the identity pool (from the ADIdentity PowerShell snap-in) that the scheme uses.

IdentityPoolName

The name of the identity pool (from the ADIdentity PowerShell snap-in) that the scheme uses.

## HostingUnitUid

The unique identifier of the hosting unit (from the Hosting Unit PowerShell snap-in) that the new provisioning scheme will use.

## HostingUnitName

The name of the hosting unit (from the Hosting Unit PowerShell snap-in) that the new provisioning scheme will use.

## CleanOnBoot

Indicates whether the VMs created are to be reset to a clean state on each boot.

## TaskId

The identifier of any current task that is running for the provisioning scheme.

## Notes

In the case of failure, the following errors can result.

## Error Codes

-----

### ProvisioningSchemeNotFound

The specified provisioning scheme could not be located.

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

### CommunicationError

An error occurred while communicating with the service.

### PermissionDenied

The user does not have administrative rights to perform this operation.

### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

### HostingUnitNotFound

The hosting unit referenced by the provisioning scheme could not be resolved

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Set-ProvScheme -ProvisioningSchemeName MyScheme -VMCpuCount 2
```

Updates a provisioning scheme called "MyScheme" to use two processors on the VMs that are created from the provisioning scheme.

# Set-ProvSchemeMetadata

Jan 04, 2017

Adds or updates metadata on the given ProvisioningScheme.

## Syntax

```
Set-ProvSchemeMetadata [-ProvisioningSchemeUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ProvSchemeMetadata [-ProvisioningSchemeUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-ProvSchemeMetadata [-ProvisioningSchemeName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ProvSchemeMetadata [-ProvisioningSchemeName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-ProvSchemeMetadata [-InputObject] <ProvisioningScheme[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ProvSchemeMetadata [-InputObject] <ProvisioningScheme[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given ProvisioningScheme objects.

## Related topics

[Add-ProvSchemeMetadata](#)

[Remove-ProvSchemeMetadata](#)

## Parameters

**-ProvisioningSchemeUid**<Guid>

Id of the ProvisioningScheme

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-ProvisioningSchemeName**<String>

Name of the ProvisioningScheme

Required?	true
Default Value	

Accept Pipeline Input?	true (ByValue, ByPropertyName)
------------------------	--------------------------------

**-InputObject**<ProvisioningScheme[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the ProvisioningScheme specified. The property cannot contain any of the following characters \/:;#.\*?=<> | [] () ""

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Value**<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

System.Collections.Generic.Dictionary[String,String]

Set-ProvSchemeMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-ProvSchemeMetadata -ProvisioningSchemeUid 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	-----
property	value

Add metadata with a name of 'property' and a value of 'value' to the ProvisioningScheme with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.



# Set-ProvServiceConfigurationData

Jan 04, 2017

Sets the value for the given key in the service configuration data.

## Syntax

```
Set-ProvServiceConfigurationData [-Name] <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored for the MachineCreation Service.

## Related topics

[Remove-ProvServiceConfigurationData](#)

[Get-ProvServiceConfigurationData](#)

## Parameters

**-Name**<String>

Specifies the key name of the metadata to be added. The key must be unique.

The Name cannot contain any of the following characters \/:#.\*?=<> | []()''''

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Value**<String>

Specifies the value for the name. If the name already exists, its value is updated.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create

high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

Citrix.MachineCreation.Sdk.ServiceConfigurationData

Set-ProvServiceConfigurationData returns an object containing the new definition of the configuration.

Name <string>

Specifies the name for the item of data.

Value <string>

Specifies the value of the data.

Notes

In the case of failure the following errors can be produced.

Error Codes

-----

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

## CommunicationError

An error occurred while communicating with the service.

## ExceptionThrown

An unexpected error occurred. To locate more details see the Windows event logs on the controller being used, or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS>Set-ProvServiceConfigurationData -Name "customProperty1" -Value "value2"
```

Name	Value
-----	-----
customProperty1	value2

Set data with a name of 'customProperty1' and value of 'value2' to the service configuration.

# Set-ProvServiceMetadata

Jan 04, 2017

Adds or updates metadata on the given Service.

## Syntax

```
Set-ProvServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Allows you to store additional custom data against given Service objects.

## Related topics

[Remove-ProvServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<Service[]>

Objects to which metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Service specified. The property cannot contain any of the following characters \/:#.\*?=<>|[]()"

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Value**<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	

Accept Pipeline Input?	false
------------------------	-------

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

System.Collections.Generic.Dictionary[String,String]

Set-ProvServiceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

**Notes**

If the command fails, the following errors can be returned.

**Error Codes**

-----

**InvalidParameterCombination**

The cmdlet parameters are inconsistent.

**UnknownObject**

One of the specified objects was not found.

**DatabaseError**

An error occurred in the service while attempting a database operation.

**DatabaseNotConfigured**

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Set-ProvServiceMetadata -ServiceHostId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Service with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Set-ProvTaskMetadata

Jan 04, 2017

Adds or updates metadata on the given Task.

## Syntax

```
Set-ProvTaskMetadata [-TaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ProvTaskMetadata [-TaskId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ProvTaskMetadata [-InputObject] <Task[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ProvTaskMetadata [-InputObject] <Task[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Use this cmdlet to store additional custom data against given Task objects.

## Related topics

[Add-ProvTaskMetadata](#)

[Remove-ProvTaskMetadata](#)

[Get-ProvTask](#)

[Stop-ProvTask](#)

[Remove-ProvTask](#)

[Switch-ProvTask](#)

## Parameters

**-TaskId**<Guid>

Id of the Task

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<Task[]>



Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Task specified. The property cannot contain any of the following characters \/:;#.\*?=<>|[]()"

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Value**<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

System.Collections.Generic.Dictionary[String,String]

Set-ProvTaskMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

### Notes

If the command fails, the following errors can result.

Error Codes

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Set-ProvTaskMetadata -TaskId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Task with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Stop-ProvTask

Jan 04, 2017

Stops currently running MachineCreation Service tasks.

## Syntax

```
Stop-ProvTask [-TaskId] <Guid> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables tasks currently running in the MachineCreation Service to be stopped. Once stopped, tasks cannot be restarted.

## Related topics

[Get-ProvTask](#)

[Remove-ProvTask](#)

[Switch-ProvTask](#)

## Parameters

**-TaskId**<Guid>

Specifies the identifier for the task to be stopped.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Input Type

System.Management.Automation.PSObject Objects containing the TaskId parameter can be piped to the Stop-ProvTask command.

### Notes

If the command fails, the following errors can result.

#### Error Codes

-----

#### InvalidWorkflow

The specified task could not be found.

#### InvalidWorkflowHost

The specified task is executing on a different server.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

DatabaseError

There was a problem communicating with the database.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ProvTask -active $true | Stop-ProvTask
```

Success

Terminate all tasks currently executing on the MachineCreation Service.

----- **EXAMPLE 2** -----

```
c:\PS>Stop-ProvTask -TaskId bd52e688-e40d-4790-83de-9f7633481454
```

Success

Terminate the named task executing on the MachineCreation Service.

# Switch-ProvTask

Jan 04, 2017

Moves all MachineCreation Service tasks from the current execution host to another.

## Syntax

```
Switch-ProvTask [-Host2] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables tasks running within the MachineCreation Service to be moved from one execution host to a different host.

Tasks can only execute on a single host. If the host is removed from a deployment, the tasks cannot continue to execute. These 'orphaned' tasks can be moved to a different host within the deployment so that they can continue to execute. All tasks must be moved; there is no mechanism to move individual tasks. Run the Switch-ProvTask command against the host to which the tasks are to be moved; that is, if you are not running the command directly on the host, use the AdminAddress parameter to specify the host to which tasks will be moved.

## Related topics

[Get-ProvTask](#)

[Stop-ProvTask](#)

[Remove-ProvTask](#)

## Parameters

**-Host2**<String>

Specifies the host from which the tasks should be removed.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	

Accept Pipeline Input?	false
------------------------	-------

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Notes

If the command fails, the following errors can result.

Error Codes

-----

PartialData

Only a subset of the requested data was returned.

NoOp

The operation was successful but had no effect.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.



ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Switch-ProvTask -Host CRASHED
```

Success

Transfer the execution of tasks that had been executing on the MachineCreation Service instance on host CRASHED to the local instance.

# Test-ProvDBConnection

Jan 04, 2017

Tests a database connection for the MachineCreation Service.

## Syntax

```
Test-ProvDBConnection [-DBConnection] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Tests a connection to the database in which the MachineCreation Service can store its state. The service will attempt to connect to the database without affecting the current connection to the database.

You do not have to clear the connection to use this command.

## Related topics

[Get-ProvServiceStatus](#)

[Get-ProvDBConnection](#)

[Set-ProvDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be tested by the MachineCreation Service.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.

Accept Pipeline Input?	false
------------------------	-------

## Return Values

### Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Test-ProvDBConnection command returns an object containing the status of the MachineCreation Service if the connection string of the specified data store were to be set to the string being tested, together with extra diagnostics information for the specified connection string.

### DBRejectedConnection

The database rejected the logon attempt from the MachineCreation Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

### InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the MachineCreation Service schema has not been added to the database.

### DBNotFound

The specified database could not be located with the configured connection string.

### DBNewerVersionThanService

The version of the MachineCreation Service currently in use is incompatible with the version of the MachineCreation Service schema on the database. Upgrade the MachineCreation Service to a more recent version.

### DBOlderVersionThanService

The version of the MachineCreation Service schema on the database is incompatible with the version of the MachineCreation Service currently in use. Upgrade the database schema to a more recent version.

### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

### OK

The Set-ProvDBConnection command would succeed if it were executed with the supplied connection string.

### Failed

The MachineCreation Service has failed.

### Unknown

The status of the MachineCreation Service cannot be determined.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidDBConnectionString

The database connection string has an invalid format.

#### PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Test-ProvDBCConnection -DBCConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Tests a database connection string for the MachineCreation Service.

----- **EXAMPLE 2** -----

```
c:\PS>Test-ProvDBCConnection -DBCConnection "Invalid Connection String"
```

Invalid database connection string format.

Tests an invalid database connection string for the MachineCreation Service.

# Test-ProvSchemeNameAvailable

Jan 04, 2017

Checks to ensure that the proposed name for a provisioning scheme is unused.

## Syntax

```
Test-ProvSchemeNameAvailable -ProvisioningSchemeName <String[]> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Checks to ensure that the proposed name for a provisioning scheme is unused. This check is done without regard for scoping of existing provisioning schemes, so the names of inaccessible schemes are also checked.

## Related topics

[New-ProvScheme](#)

[Rename-ProvScheme](#)

## Parameters

**-ProvisioningSchemeName**<String[]>

The name or names of the provisioning scheme(s) to be tested.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

NameAvailability

Pairs of name and the availability of the name

## Notes

In the case of failure, the following errors can result.

### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### CommunicationError

An error occurred while communicating with the service.

#### PermissionDenied

The user does not have administrative rights to perform this operation.

#### ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

Test-ProvSchemeNameAvailable -ProvisioningSchemeName \$NewSchemeName

This tests whether the value of \$NewSchemeName is unique or not, and can be used to create a new provisioning scheme or rename an existing one without failing. True is returned if the name is good.

# Unlock-ProvScheme

Jan 04, 2017

Unlocks a Provisioning Scheme.

## Syntax

```
Unlock-ProvScheme [-ProvisioningSchemeName] <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Unlock-ProvScheme -ProvisioningSchemeUid <Guid> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability to unlock a provisioning scheme that has the Id of a failed Task still associated with it. This allows another long-running task to operate on that scheme. The cmdlet will not unlock a scheme with a task still marked as being active. Use Stop-ProvTask (or, if the task is registered with a failed server, Switch-ProvTask and Stop-ProvTask) to stop any active task first.

## Related topics

[Get-ProvScheme](#)

[Stop-ProvTask](#)

## Parameters

**-ProvisioningSchemeName**<String>

The name of the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ProvisioningSchemeUid**<Guid>

The unique identifier of the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	LocalHost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Notes

In the case of failure, the following errors can result.

#### Error Codes

-----

#### ProvisioningSchemeNotFound

The specified provisioning scheme could not be located.

#### NoOp

The specified provisioning scheme had no task object associated.

#### TaskActive

The task object associated with the provisioning scheme is still active.

#### DatabaseError

An error occurred in the service while attempting a database operation.



CouldNotQueryDatabase

The query required to get the database was not defined.

CommunicationError

An error occurred while communicating with the service.

InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

PermissionDenied

The user does not have administrative rights to perform this operation.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
C:\PS>Unlock-ProvScheme -provisioningSchemeName MyScheme  
Unlocks the provisioning scheme 'MyScheme'.
```

# Unlock-ProvVM

Jan 04, 2017

Unlocks a VM.

## Syntax

```
Unlock-ProvVM [-VMID] <String[]> -ProvisioningSchemeName <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Unlock-ProvVM [-VMID] <String[]> -ProvisioningSchemeUid <Guid> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability to unlock a virtual machine.

## Related topics

[Get-ProvVM](#)

[Lock-ProvVM](#)

## Parameters

**-VMID**<String[]>

The virtual machine Id (hypervisor context)

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ProvisioningSchemeName**<String>

The name of the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-ProvisioningSchemeUid**<Guid>

The unique identifier of the provisioning scheme.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller that the PowerShell snap-in connects to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Input Type**

Citrix.MachineCreation.Sdk.ProvisionedVirtualMachine You can pipe an object containing a parameter called 'VMId' and 'ProvisioningSchemeName' to Lock-ProvVM

**Notes**

In the case of failure, the following errors can result.

**Error Codes**

-----

VMDoesNotExist

The specified VM cannot be located.

VMAAlreadyUnLocked

The VM is already unlocked.

VMDoesNotExistForProvisioningScheme

The specified VM does exist in the hypervisor, but is not part of the specified provisioning scheme.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

CommunicationError

An error occurred while communicating with the service.

PermissionDenied

The user does not have administrative rights to perform this operation.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error

ExceptionThrown

An unexpected error occurred. To locate more details, see the Windows event logs on the controller being used or examine the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS>Unlock-ProvVM -provisioningSchemeName MyScheme -VMId bc79802c-ba6e-8de8-99e9-4c35d7ad24b4
Unlocks the VM with the Id 'bc79802c-ba6e-8de8-99e9-4c35d7ad24b4' in the provisioning scheme 'MyScheme'.
```

### ----- EXAMPLE 2 -----

```
C:\PS>Get-ProvVM -provisioningSchemeName MyScheme | Unlock-ProvVM
Unlocks all the VMs in the provisioning scheme 'MyScheme'.
```

### ----- EXAMPLE 3 -----

```
C:\PS>Get-ProvVM -Locked $True | Unlock-ProvVM
```

Unlocks all the VMs that are currently locked.

# Citrix.Monitor.Admin.V1

Jan 04, 2017  
Overview

Name	Description
<a href="#">MonitorMonitorSnapin</a>	The Monitor Service PowerShell snap-in provides administrative functions for the Monitor Service.
<a href="#">Monitor Filtering</a>	Describes the common filtering options for XenDesktop cmdlets.

## Cmdlets

Name	Description
<a href="#">Get-MonitorConfiguration</a>	Gets the configuration settings currently being used by the Monitor Service.
<a href="#">Get-MonitorDataStore</a>	Gets details for each of the Monitor data stores.
<a href="#">Get-MonitorDBConnection</a>	Gets the database string for the specified data store used by the Monitor Service.
<a href="#">Get-MonitorDBSchema</a>	Gets a script that creates the Monitor Service database schema for the specified data store.
<a href="#">Get-MonitorDBVersionChangeScript</a>	Gets a script that updates the Monitor Service database schema.
<a href="#">Get-MonitorInstalledDBVersion</a>	Gets a list of all available database schema versions for the Monitor Service.
<a href="#">Get-MonitorService</a>	Gets the service record entries for the Monitor Service.
<a href="#">Get-MonitorServiceAddedCapability</a>	Gets any added capabilities for the Monitor Service on the controller.
<a href="#">Get-MonitorServiceInstance</a>	Gets the service instance entries for the Monitor Service.
<a href="#">Get-MonitorServiceStatus</a>	Gets the current status of the Monitor Service on the controller.
<a href="#">Remove-MonitorServiceMetadata</a>	Removes metadata from the given Service.
<a href="#">Reset-MonitorDataStore</a>	Refreshes the database string currently being used by the Monitor service.

Name	Description
<del>Reset</del> <a href="#">MonitorServiceGroupMembership</a>	Resets the access permissions and configuration service locations for the Monitor Service.
<a href="#">Set-MonitorConfiguration</a>	Sets configuration settings that are used by the Monitor Service.
<a href="#">Set-MonitorDBConnection</a>	Configures a database connection for the Monitor Service.
<a href="#">Set-MonitorServiceMetadata</a>	Adds or updates metadata on the given Service.
<a href="#">Test-MonitorDBConnection</a>	Tests a database connection for the Monitor Service.

# about\_MonitorMonitorSnapin

Jan 04, 2017

## TOPIC

about\_MonitorMonitorSnapin

## SHORT DESCRIPTION

The Monitor Service PowerShell snap-in provides administrative functions for the Monitor Service.

## COMMAND PREFIX

All commands in this snap-in have the noun prefixed with 'Monitor'.

## LONG DESCRIPTION

The Monitor Service PowerShell snap-in enables both local and remote administration of the Monitor service. It provides facilities to query the Monitor service configuration settings and to modify those settings. It also provides the standard set of XenDesktop service cmdlets.



# about\_Monitor\_Filtering

Jan 04, 2017

## TOPIC

XenDesktop - Advanced Dataset Filtering

## SHORT DESCRIPTION

Describes the common filtering options for XenDesktop cmdlets.

## LONG DESCRIPTION

Some cmdlets operate on large quantities of data and, to reduce the overhead of sending all of that data over the network, many of the Get- cmdlets support server-side filtering of the results.

The conventional way of filtering results in PowerShell is to pipeline them into Where-Object, Select-Object, and Sort-Object, for example:

```
Get-<Noun> | Where { $_.Size = 'Small' } | Sort 'Date' | Select -First 10
```

However, for most XenDesktop cmdlets the data is stored remotely and it would be slow and inefficient to retrieve large amounts of data over the network and then discard most of it. Instead, many of the Get- cmdlets provide filtering parameters that allow results to be processed on the server, returning only the required results.

You can filter results by most object properties using parameters derived from the property name. You can also sort results or limit them to a specified number of records:

```
Get-<Noun> -Size 'Small' -SortBy 'Date' -MaxRecordCount 10
```

You can express more complex filter conditions using a syntax and set of operators very similar to those used by PowerShell expressions.

Those cmdlets that support filtering have the following common parameters:

-MaxRecordCount <int>

Specifies the maximum number of results to return.  
For example, to return only the first nine results use:

```
Get-<Noun> -MaxRecordCount 9
```

If not specified, only the first 250 records are returned, and if more are available, a warning is produced:

WARNING: Only first 250 records returned. Use -MaxRecordCount to retrieve more.

You can suppress this warning by using -WarningAction or by specifying a value for -MaxRecordCount.

To retrieve all records, specify a large number for -MaxRecordCount. As the value is an integer, you can use the following:

```
Get-<Noun> -MaxRecordCount [int]::MaxValue
```

-ReturnTotalRecordCount [<SwitchParameter>]

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. For example:

```
Get-<Noun> -MaxRecordCount 9 -ReturnTotalRecordCount  
....
```

```
Get-<Noun> : Returned 9 of 10 items  
At line:1 char:18  
+ Get-<Noun> <<<< -MaxRecordCount 9 -ReturnTotalRecordCount  
+ CategoryInfo          : OperationStopped: (:) [Get-<Noun>], PartialDataException  
+ FullyQualifiedErrorId : PartialData,Citrix.<SDKName>.SDK.Get<Noun>
```

The count can be accessed using the TotalAvailableResultCount property:

```
$count = $error[0].TotalAvailableResultCount
```

-Skip <int>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

-SortBy <string>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order, respectively. Ascending order is assumed if no prefix is present.

Sorting occurs before -MaxRecordCount and -Skip parameters are applied. For example, to sort by Name and then by Count (largest first) use:

```
-SortBy 'Name,-Count'
```

By default, sorting by an enumeration property uses the numeric value of the elements. You can specify a different sort order by qualifying the name with an ordered list of elements or their numeric values, or <null> to indicate the placement of null values.

Elements not mentioned are placed at the end in their numeric order. For example, to sort by two different enums and then by the object id:

```
-SortBy 'MyState(StateC,<null>,StateA,StateB),Another(0,3,2,1),Id'
```

-Filter <String>

This parameter lets you specify advanced filter expressions, and supports combination of conditions with -and and -or, and grouping with braces. For example:

```
Get-<Noun> -Filter 'Name -like "High*" -or (Priority -eq 1 -and Severity -ge 2)'
```

The syntax is close enough to PowerShell syntax that you can use script blocks in most cases. This can be easier to read as it reduces quoting:

```
Get-<Noun> -Filter { Count -ne $null }
```

The full -Filter syntax is provided below.

## EXAMPLES

Filtering by strings performs a case-insensitive wildcard match.

Separate parameters are combined with an implicit -and operator.

Normal PowerShell quoting rules apply, so you can use single or double quotes, and omit the quotes altogether for many strings. The order of parameters does not make any difference. The following are equivalent:

```
Get-<Noun> -Company Citrix -Product Xen*
Get-<Noun> -Company "citrix" -Product '[X]EN*'
Get-<Noun> -Product "Xen*" -Company "CITRIX"
Get-<Noun> -Filter { Company -eq 'Citrix' -and Product -like 'Xen*' }
```

See [about\\_Quoting\\_Rules](#) and [about\\_Wildcards](#) for details about PowerShell handling of quotes and wildcards.

To avoid wildcard matching or include quote characters, you can escape the wildcards using the normal PowerShell escape mechanisms (see [about\\_Escape\\_Characters](#)), or switch to a filter expression and the `-eq` operator:

```
Get-<Noun> -Company "Abc[*]"           # Matches Abc*
Get-<Noun> -Company "Abc`*"           # Matches Abc*
Get-<Noun> -Filter { Company -eq "Abc*" } # Matches Abc*
Get-<Noun> -Filter { Company -eq "A`"B`C" } # Matches A"B'C
```

Simple filtering by numbers, booleans, and TimeSpans perform direct equality comparisons, although if the value is nullable you can also search for null values. Here are some examples:

```
Get-<Noun> -Uid 123
Get-<Noun> -Enabled $true
Get-<Noun> -Duration 1:30:40
Get-<Noun> -NullableProperty $null
```

More comparisons are possible using advanced filtering with `-Filter`:

```
Get-<Noun> -Filter 'Capacity -ge 10gb'
Get-<Noun> -Filter 'Age -ge 20 -and Age -lt 40'
Get-<Noun> -Filter 'VolumeLevel -like "[123]"'
Get-<Noun> -Filter 'Enabled -ne $false'
Get-<Noun> -Filter 'NullableProperty -ne $null'
```

You can check boolean values without an explicit comparison operator, and you can also combine them with `-not`:

```
Get-<Noun> -Filter 'Enabled' # Equivalent to 'Enabled -eq $true'
Get-<Noun> -Filter '-not Enabled' # Equivalent to 'Enabled -eq $false'
```

See [about\\_Comparison\\_Operators](#) for an explanation of the operators, but note that only a subset of PowerShell operators are supported (`-eq`, `-ne`, `-gt`, `-ge`, `-lt`, `-le`, `-like`, `-notlike`, `-in`, `-notin`, `-contains`, `-notcontains`).

Enumeration values can either be specified using typed values or the string name of the enumeration value:

```
Get-<Noun> -Shape [Shapes]::Square
Get-<Noun> -Shape Circle
```

With filter expressions, typed values can be specified with simple variables or quoted strings. They also support enumerations with wildcards:

```
$s = [Shapes]::Square
Get-<Noun> -Filter { Shape -eq $s -or Shape -eq "Circle" }
Get-<Noun> -Filter { Shape -like 'C*' }
```

By their nature, floating point values, DateTime values, and TimeSpan values are best suited to relative comparisons rather than just equality. DateTime strings are converted using the locale and time zone of the user device, but you can use ISO8601 format strings (YYYY-MM-DDThh:mm:ss.sTZD) to avoid ambiguity. You can also use standard PowerShell syntax to create these values:

```
Get-<Noun> -Filter { StartTime -ge "2010-08-23T12:30:00.0Z" }
$d = [DateTime]"2010-08-23T12:30:00.0Z"
Get-<Noun> -Filter { StartTime -ge $d }
$d = (Get-Date).AddDays(-1)
Get-<Noun> -Filter { StartTime -ge $d }
```

Relative times are quite common and, when using filter expressions, you can also specify DateTime values using a relative format:

```
Get-<Noun> -Filter { StartTime -ge '-2' } # Two days ago
Get-<Noun> -Filter { StartTime -ge '-1:30' } # Hour and a half ago
Get-<Noun> -Filter { StartTime -ge '-0:0:30' } # 30 seconds ago
```

## ARRAY PROPERTIES

When filtering against list or array properties, simple parameters perform a case-insensitive wildcard match against each of the members. With filter expressions, you can use the -contains and -notcontains operators. Unlike PowerShell, these perform wildcard matching on strings.

Note that for array properties the naming convention is for the returned property to be plural, but the parameter used to search for any match is singular. The following are equivalent (assuming Users is an array property):

```
Get-<Noun> -User Fred*
Get-<Noun> -Filter { User -like "Fred*" }
Get-<Noun> -Filter { Users -contains "Fred*" }
```

You can also use the singular form with -Filter to search using other operators:

```
# Match if any user in the list is called "Frederick"
Get-<Noun> -Filter { User -eq "Frederick" }
# Match if any user in the list has a name alphabetically below 'F'
Get-<Noun> -Filter { User -lt 'F' }
```

## COMPLEX EXPRESSIONS

When matching against multiple values, you can use a sequence of comparisons joined with -or operators, or you can use -in and -notin:

```
Get-<Noun> -Filter { Shape -eq 'Circle' -or Shape -eq 'Square' }
$shapes = 'Circle','Square'
Get-<Noun> -Filter { Shape -in $shapes }
$sides = 1..4
Get-<Noun> -Filter { Sides -notin $sides }
```

Braces can be used to group complex expressions, and override the default left-to-right evaluation of -and and -or. You can also use -not to invert the sense of any sub-expression:

```
Get-<Noun> -Filter { Size -gt 4 -or (Color -eq 'Blue' -and Shape -eq 'Circle') }
Get-<Noun> -Filter { Sides -lt 5 -and -not (Color -eq 'Blue' -and Shape -eq 'Circle') }
```

## PAGING

The simplest way to page through data is to use the -Skip and -MaxRecordCount parameters. So, to read the first three pages of data with 10 records per page, use:

```
Get-<Noun> -Skip 0 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 10 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 20 -MaxRecordCount 10 <other filtering criteria>
```

You must include the same filtering criteria on each call, and ensure that the data is sorted consistently.

The above approach is often acceptable, but as each call performs an independent query, data changes can result in records being skipped or appearing twice. One approach to improve this is to sort by a unique id field and then start the search for the next page at the unique id after the last unique id of the previous page. For example:

```
# Get the first page
Get-<Noun> -MaxRecordCount 10 -SortBy SerialNumber

SerialNumber ...
----- ---
A120004
A120007
... 7 other records ...
A120900

# Get the next page
Get-<Noun> -MaxRecordCount 10 -Filter { FirstName -gt 'A120900' }

SerialNumber ...
```

----- ---  
A120901  
B220000  
...

## FILTER SYNTAX DEFINITION

<Filter> ::= <ScriptBlock> | <ComponentList>

<ScriptBlock> ::= "{" <ComponentList> "}"

<ComponentList> ::= <Component> <AndOrOperator> <ComponentList> |

<Component>

<Component> ::= <NotOperator> <Factor> |

<Factor>

<Factor> ::= "(" <ComponentList> ")" |

<PropertyName> <ComparisonOperator> <Value> |  
<PropertyName>

<AndOrOperator> ::= "-and" | "-or"

<NotOperator> ::= "-not" | "!"

<ComparisonOperator>

::= "-eq" | "-ne" | "-le" | "-ge" | "-lt" | "-gt" |  
"-like" | "-notlike" | "-contains" | "-notcontains" |  
"-in" | "-notin"

<PropertyName> ::= <simple name of property>

<Value> ::= <string literal> | <numeric literal> |

<scalar variable> | <array variable> |  
"\$null" | "\$true" | "\$false"

Numeric literals support decimal and hexadecimal literals, with optional multiplier suffixes (kb, mb, gb, tb, pb).

Dates and times can be specified as string literals. The current culture determines what formats are accepted. To avoid any ambiguity, use strings formatted to the ISO8601 standard. If not specified, the current time zone is used.

Relative date-time string literals are also supported, using a minus sign followed by a TimeSpan. For example, "-1:30" means 1 hour and 30 minutes ago.



# Get-MonitorConfiguration

Jan 04, 2017

Gets the configuration settings currently being used by the Monitor Service.

## Syntax

```
Get-MonitorConfiguration [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the configuration currently being used by the Monitor Service.

A site database connection must be configured for this command to be used.

## Related topics

[Set-MonitorConfiguration](#)

## Parameters

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Get-MonitorConfiguration returns the configuration settings.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-MonitorConfiguration
```

Gets the current configuration for the Monitor Service.

# Get-MonitorDataStore

Jan 04, 2017

Gets details for each of the Monitor data stores.

## Syntax

```
Get-MonitorDataStore [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns an object for each of the Monitor data stores describing the connection string, data store name, db type, provider, schema name, and DB status.

A database connection must be configured in order for this command to be used if the service has a secondary data store. This is not required for the site data store.

## Related topics

[Reset-MonitorDataStore](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.Monitor.Sdk.DataStoreConfiguration

An object describing the connection string, data store name, database type, provider, schema name and database status.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

### DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-MonitorDataStore
```

```
ConnectionString : Server=.\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True
DataStore      : Site
DatabaseType   : SqlServer
Provider       : MSSQL
SchemaName     : MonitorSiteSchema
Status         : OK
```

```
ConnectionString :
DataStore       : Secondary
DatabaseType    : SqlServer
Provider        : MSSQL
SchemaName      : MonitorSecondarySchema
Status          : DBUnconfigured
```

Get the database connection string for the Monitor Service.

# Get-MonitorDBConnection

Jan 04, 2017

Gets the database string for the specified data store used by the Monitor Service.

## Syntax

```
Get-MonitorDBConnection [-DataStore] <String> [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the database connection string for the specified data store.

If the returned string is blank, no valid connection string has been specified. In this case the service is running, but is idle and awaiting specification of a valid connection string.

## Related topics

[Get-MonitorServiceStatus](#)

[Get-MonitorDataStore](#)

[Set-MonitorDBConnection](#)

[Test-MonitorDBConnection](#)

## Parameters

**-DataStore**<String>

Specifies the logical name of the data store for the Monitor Service. Can be either be 'Site' or the logical name of the secondary data store.

Required?	false
Default Value	Site
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.

Accept Pipeline Input?	false
------------------------	-------

## Return Values

system.string

The database connection string configured for the Monitor Service.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoDBConnections

The database connection string for the Monitor Service has not been specified.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-MonitorDBConnection
```

Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True  
Get the database connection string for the Monitor Service.

# Get-MonitorDBSchema

Jan 04, 2017

Gets a script that creates the Monitor Service database schema for the specified data store.

## Syntax

```
Get-MonitorDBSchema [-DatabaseName <String>] [-ServiceGroupName <String>] [-ScriptType <ScriptTypes>] [-LocalDatabase] [-Sid <String>] [-DataStore <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets SQL scripts that can be used to create a new Monitor Service database schema, add a new Monitor Service to an existing site, remove a Monitor Service from a site, or create a database server logon for a Monitor Service. If no Sid parameter is provided, the scripts obtained relate to the currently selected Monitor Service instance, otherwise the scripts relate to Monitor Service instance running on the machine identified by the Sid provided. When obtaining the Evict script, a Sid parameter must be supplied. The current service instance is that on the local machine, or that explicitly specified by the last usage of the -AdminAddress parameter to a Monitor SDK cmdlet. The service instance used to obtain the scripts does not need to be a member of a site or to have had its database connection configured. The database scripts support only Microsoft SQL Server, or SQL Server Express, and require Windows integrated authentication to be used. They can be run using SQL Server's SQLCMD utility, or by copying the script into an SQL Server Management Studio (SSMS) query window and executing the query. If using SSMS, the query must be executed in 'SMDCMD mode'. The ScriptType parameter determines which script is obtained. If ScriptType is not specified, or is FullDatabase, the script contains:

- o Creation of service schema
- o Creation of database server logon
- o Creation of database user
- o Addition of database user to Monitor Service roles

If ScriptType is Instance, the returned script contains:

- o Creation of database server logon
- o Creation of database user
- o Addition of database user to Monitor Service roles

If ScriptType is Evict, the returned script contains:

- o Removal of Monitor Service instance from database
- o Removal of database user

If ScriptType is Login, the returned script contains:

- o Creation of database server logon only

If the service uses two data stores they can exist in the same database. You do not need to configure a database before using this command.

## Related topics

[Get-MonitorDataStore](#)

[Set-MonitorDBConnection](#)



## Parameters

### **-DatabaseName**<String>

Specifies the name of the database for which the schema will be generated.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-ServiceGroupName**<String>

Specifies the name of the service group to be used when creating the database schema. The service group is a collection of all the Monitor services that share the same database instance and are considered equivalent; that is, all the services within a service group can be used interchangeably.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-ScriptType**<ScriptTypes>

Specifies the type of database script returned. Available script types are:

#### Database

Returns a full database script that can be used to create a database schema for the Monitor Service in a database instance that does not already contain a schema for this service. The DatabaseName and ServiceGroupName parameters must be specified to create a script of this type.

#### Instance

Returns a permissions script that can be used to add further Monitor services to an existing database instance that already contains the full Monitor service schema, associating the services to the Service Group. The Sid parameter can optionally be specified to create a script of this type.

#### Login

Returns a database logon script that can be used to add the required logon accounts to an existing database instance that contains the Monitor Service schema. This is used primarily when creating a mirrored database environment. The DatabaseName parameter must be specified to create a script of this type.

#### Evict

Returns a script that can be used to remove the specified Monitor Service from the database entirely. The DatabaseName and Sid parameters must be specified to create a script of this type.

Required?	false
-----------	-------

Default Value	Database
Accept Pipeline Input?	false

**-LocalDatabase**<SwitchParameter>

Specifies whether the database script is to be used in a database instance run on the same controller as other services in the service group. Including this parameter ensures the script creates only the required permissions for local services to access the database schema for Monitor services.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Sid**<String>

Specifies the SID of the controller on which the Monitor Service instance to remove from the database is running.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByValue)

**-DataStore**<String>

Specifies the logical name of the data store for the Monitor Service. Can be either be 'Site' or the logical name of the secondary data store.

Required?	false
Default Value	Site
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.

Accept Pipeline Input?	false
------------------------	-------

## Return Values

System.string

A string containing the required SQL script for application to a database.

## Notes

The scripts returned support Microsoft SQL Server Express Edition, Microsoft SQL Server Standard Edition, and Microsoft SQL Server Enterprise Edition databases only, and are generated on the assumption that integrated authentication will be used.

If the ScriptType parameter is not included or set to 'FullDatabase', the full database script is returned, which will:

Create the database schema.

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist).

If the ScriptType parameter is set to 'Instance', the script will:

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist) and associate it with a user.

If the ScriptType parameter is set to 'Login', the script will:

Create the logon (providing the schema does not already exist) and associate it with a pre-existing user of the same name.

If the LocalDatabase parameter is included, the NetworkService account will be added to the list of accounts permitted to access the database. This is required only if the database is run on a controller.

If the command fails, the following errors can be returned.

## Error Codes

-----

### GetSchemasFailed

The database schema could not be found.

### ActiveDirectoryAccountResolutionFailed

The specified Active Directory account or Group could not be found.

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

### PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-MonitorDBSchema -DatabaseName MyDB -ServiceGroupName MyServiceGroup > c:\MonitorSchema.sql
```

Get the full database schema for site data store of the Monitor Service and copy it to a file called 'c:\MonitorSchema.sql'.

This script can then be used to create the schema in a pre-existing database named 'MyDB' that does not already contain a Monitor Service site schema.

----- **EXAMPLE 2** -----

```
c:\PS>Get-MonitorDBSchema -DatabaseName MyDB -scriptType Login > c:\MonitorLogins.sql
```

Get the logon scripts for the Monitor Service.

----- **EXAMPLE 3** -----

```
c:\PS>Get-MonitorDBSchema -DatabaseName MyDB -ServiceGroupName MyServiceGroup -DataStore Secondary > c:\MonitorSchema.sql
```

Get the full database schema for the secondary data store of the Monitor Service and copy it to a file called 'c:\MonitorSecondarySchema.sql'.

This script can then be used to create the schema in a pre-existing database named 'MyDB' that does not already contain a Monitor Service secondary schema.

# Get-MonitorDBVersionChangeScript

Jan 04, 2017

Gets a script that updates the Monitor Service database schema.

## Syntax

```
Get-MonitorDBVersionChangeScript -DatabaseName <String> -TargetVersion <Version> [-DataStore <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns a database script that can be used to upgrade or downgrade the site or secondary schema for the Monitor Service from the current schema version to a different version.

## Related topics

[Get-MonitorInstalledDBVersion](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database instance to which the update applies.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-TargetVersion**<Version>

Specifies the version of the database you want to update to.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-DataStore**<String>

Specifies the logical name of the data store for the Monitor Service. Can be either be 'Site' or the logical name of the secondary data store.

Required?	false
Default Value	Site
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

System.Management.Automation.PSObject

A PSObject containing the required SQL script for application to a database.

### Notes

The PSObject returned by this cmdlet contains the following properties:

- Script The raw text of the SQL script to apply the update, or null in the case when no upgrade path to the specified target version exists.
- NeedExclusiveAccess Indicates whether all services in the service group must be shut down during the update or not.
- CanUndo Indicates whether the generated script allows the updated schema to be reverted to the state prior to the update.

Scripts to update the schema version are stored in the database so any service in the service group can obtain these scripts. Extreme caution should be exercised when using update scripts. Citrix recommends backing up the database before attempting to upgrade the schema. Database update scripts may require exclusive use of the schema and so may not be able to execute while any Monitor services are running. However, this depends on the specific update being carried out.

After a schema update has been carried out, services that require the previous version of the schema may cease to operate. The ServiceState parameter reported by the Get-MonitorServiceStatus command provides information about service compatibility. For example, if the schema has been upgraded to a more recent version that a service cannot use, the service reports "DBNewerVersionThanService".

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoOp

The operation was successful but had no effect.

#### NoDBConnections

The database connection string for the Monitor Service has not been specified.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
C:\PS> $update = Get-MonitorDBVersionChangeScript -DatabaseName MyDb -TargetVersion 1.0.75.0
```

```
C:\PS> $update.Script > update_75.sql
```

Gets an SQL update script to update the current schema to version 1.0.75.0. The resulting update\_75.sql script is suitable for direct use with the SQL Server SQLCMD utility.

# Get-MonitorInstalledDBVersion

Jan 04, 2017

Gets a list of all available database schema versions for the Monitor Service.

## Syntax

```
Get-MonitorInstalledDBVersion [-Upgrade] [-Downgrade] [-DataStore <String>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Returns the current version of the Monitor Service database schema, if no flags are set, otherwise returns versions for which upgrade or downgrade scripts are available and have been stored in the database.

## Related topics

### Parameters

#### **-Upgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be updated should be returned.

Required?	false
Default Value	
Accept Pipeline Input?	false

#### **-Downgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be reverted should be returned.

Required?	false
Default Value	
Accept Pipeline Input?	false

#### **-DataStore**<String>

Specifies the database connection logical name the schema script should be returned for. The parameter is optional.

Required?	false
Default Value	



Accept Pipeline Input?	false
------------------------	-------

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Return Values

System.Version

The Get-MonitorInstalledDbVersion command returns objects containing the new definition of the Monitor Service database schema version.

Major <Integer>

Minor <Integer>

Build <Integer>

Revision <Integer>

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

InvalidParameterCombination

Both the Upgrade and Downgrade flags were specified.

NoOp

The operation was successful but had no effect.

NoDBConnections

The database connection string for the Monitor Service has not been specified.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-MonitorInstalledDBVersion
```

```
Major Minor Build Revision
```

```
-----
```

```
5 6 0 0
```

Get the currently installed version of the Monitor Service database schema.

----- **EXAMPLE 2** -----

```
c:\PS>Get-MonitorInstalledDBVersion -Upgrade
```

```
Major Minor Build Revision
```

```
-----
```

```
6 0 0 0
```

Get the versions of the Monitor Service database schema for which upgrade scripts are supplied.

# Get-MonitorService

Jan 04, 2017

Gets the service record entries for the Monitor Service.

## Syntax

```
Get-MonitorService [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns instances of the Monitor Service that the service publishes. The service records contain account security identifier information that can be used to remove each service from the database.

A database connection for the service is required to use this command.

## Related topics

### Parameters

#### **-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: `-Metadata "abc:x"` matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

#### **-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

#### **-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See `about_Monitor_Filtering` for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by `-ReturnTotalRecordCount`.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

### **-Filter**<String>

Gets records that match a PowerShell-style filter expression. See [about\\_Monitor\\_Filtering](#) for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

Citrix.Monitor.Sdk.Service

The `Get-MonitorServiceInstance` command returns an object containing the following properties.

**Uid** <Integer>

Specifies the unique identifier for the service in the group. The unique identifier is an index number.

**ServiceHostId** <Guid>

Specifies the unique identifier for the service instance.

**DNSName** <String>

Specifies the domain name of the host on which the service runs.

**MachineName** <String>

Specifies the short name of the host on which the service runs.

**CurrentState** <Citrix.Fma.Sdk.ServiceCore.ServiceState>

Specifies whether the service is running, started but inactive, stopped, or failed.

LastStartTime <DateTime>

Specifies the date and time at which the service was last restarted.

LastActivityTime <DateTime>

Specifies the date and time at which the service was last stopped or restarted.

OSType

Specifies the operating system installed on the host on which the service runs.

OSVersion

Specifies the version of the operating system installed on the host on which the service runs.

ServiceVersion

Specifies the version number of the service instance. The version number is a string that reflects the full build version of the service.

DatabaseUserName <string>

Specifies for the service instance the Active Directory account name with permissions to access the database. This will be either the machine account or, if the database is running on a controller, the NetworkService account.

Sid <string>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

ActiveSiteServices <string[]>

Specifies the names of active site services currently running in the service. Site services are components that perform long-running background processing in some services. This field is empty for services that do not contain site services.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### PartialData

Only a subset of the available data was returned.

#### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

#### CouldNotQueryDatabase

The query required to get the database was not defined.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-MonitorService
```

```
Uid          : 1
ServiceHostId : aef6f464-f1ee-4042-a523-66982e0cecd0
DNSName      : MyServer.company.com
MachineName  : MYSERVER
CurrentState  : On
LastStartTime : 04/04/2011 15:25:38
LastActivityTime : 04/04/2011 15:33:39
OSType       : Win32NT
OSVersion    : 6.1.7600.0
ServiceVersion : 5.1.0.0
DatabaseUserName : NT AUTHORITY\NETWORK SERVICE
SID          : S-1-5-21-2316621082-1546847349-2782505528-1165
ActiveSiteServices : {MySiteService1, MySiteService2...}
Get all the instances of the Monitor Service running in the current service group.
```

# Get-MonitorServiceAddedCapability

Jan 04, 2017

Gets any added capabilities for the Monitor Service on the controller.

## Syntax

```
Get-MonitorServiceAddedCapability [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables updates to the Monitor Service on the controller to be detected.

You do not need to configure a database connection before using this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

System.String

String containing added capabilities.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.



## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- **EXAMPLE 1** -----

```
c:\PS>Get-MonitorServiceAddedCapability
```

Get the added capabilities of the Monitor Service.

# Get-MonitorServiceInstance

Jan 04, 2017

Gets the service instance entries for the Monitor Service.

## Syntax

```
Get-MonitorServiceInstance [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns service interfaces published by the instance of the Monitor Service. Each instance of a service publishes multiple interfaces with distinct interface types, and each of these interfaces is represented as a ServiceInstance object. Service instances can be used to register the service with a central configuration service so that other services can use the functionality.

You do not need to configure a database connection to use this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.Monitor.Sdk.ServiceInstance

The Get-MonitorServiceInstance command returns an object containing the following properties.

ServiceGroupUid <Guid>

Specifies the unique identifier for the service group of which the service is a member.

ServiceGroupName <String>

Specifies the name of the service group of which the service is a member.

ServiceInstanceUID <Guid>

Specifies the unique identifier for registered service instances, which are service instances held by and obtained from a

central configuration service. Unregistered service instances do not have unique identifiers.

ServiceType <String>

Specifies the service instance type. For this service, the service instance type is always Monitor.

Address

Specifies the address of the service instance. The address can be used to access the service and, when registered in the central configuration service, can be used by other services to access the service.

Binding

Specifies the binding type that must be used to communicate with the service instance. In this release of XenDesktop, the binding type is always 'wcf\_HTTP\_kerb'. This indicates that the service provides a Windows Communication Foundation endpoint that uses HTTP binding with integrated authentication.

Version

Specifies the version of the service instance. The version number is used to ensure that the correct versions of the services are used for communications.

ServiceAccount <String>

Specifies the Active Directory account name for the machine on which the service instance is running. The account name is used to provide information about the permissions required for interservice communications.

ServiceAccountSid <String>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

InterfaceType <String>

Specifies the interface type. Each service can provide multiple service instances, each for a different purpose, and the interface defines the purpose. Available interfaces are:

SDK - for PowerShell operations

InterService - for operations between different services

Peer - for communications between services of the same type

Metadata <Citrix.Monitor.Sdk.Metadata[]>

The collection of metadata associated with registered service instances, which are service instances held by and obtained from a central configuration service. Metadata is not stored for unregistered service instances.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-MonitorServiceInstance
```

```
Address      : http://MyServer.com:80/Citrix/Monitor
Binding      : wcf_HTTP_kerb
InterfaceType : SDK
Metadata     :
MetadataMap  :
ServiceAccount : ENG\MyAccount$
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164
ServiceGroupName : MyServiceGroup
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000
ServiceType  : Monitor
Version      : 1
```

```
Address      : http://MyServer.com:80/Citrix/Monitor/IServiceApi
Binding      : wcf_HTTP_kerb
InterfaceType : InterService
Metadata     :
MetadataMap  :
```

ServiceAccount : ENG\MyAccount  
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164  
ServiceGroupName : MyServiceGroup  
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d  
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000  
ServiceType : Monitor  
Version : 1

Get all instances of the Monitor Service running on the specified machine. For remote services, use the AdminAddress parameter to define the service for which the interfaces are required. If the AdminAddress parameter has not been specified for the runspace, service instances running on the local machine are returned.

# Get-MonitorServiceStatus

Jan 04, 2017

Gets the current status of the Monitor Service on the controller.

## Syntax

```
Get-MonitorServiceStatus [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables the status of the Monitor Service on the controller to be monitored. If the service has multiple data stores it will return the overall state as an aggregate of all the data store states. For example, if the site data store status is OK and the secondary data store status is DBUnconfigured then it will return DBUnconfigured.

## Related topics

[Get-MonitorDataStore](#)

[Set-MonitorDBConnection](#)

[Test-MonitorDBConnection](#)

[Get-MonitorDBConnection](#)

[Get-MonitorDBSchema](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Get-MonitorServiceStatus command returns an object containing the status of the Monitor Service together with extra diagnostics information.

DBUnconfigured

The Monitor Service does not have a database connection configured.

#### DBRejectedConnection

The database rejected the logon attempt from the Monitor Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

#### InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Monitor Service schema has not been added to the database.

#### DBNotFound

The specified database could not be located with the configured connection string.

#### DBNewerVersionThanService

The version of the Monitor Service currently in use is incompatible with the version of the Monitor Service schema on the database. Upgrade the Monitor Service to a more recent version.

#### DBOlderVersionThanService

The version of the Monitor Service schema on the database is incompatible with the version of the Monitor Service currently in use. Upgrade the database schema to a more recent version.

#### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

#### OK

The Monitor Service is running and is connected to a database containing a valid schema.

#### Failed

The Monitor Service has failed.

#### Unknown

(0) The service status cannot be determined.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-MonitorServiceStatus
```

## DBUnconfigured

Get the current status of the Monitor Service.



# Remove-MonitorServiceMetadata

Jan 04, 2017

Removes metadata from the given Service.

## Syntax

```
Remove-MonitorServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-MonitorServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-MonitorServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-MonitorServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Service.

## Related topics

[Set-MonitorServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<Service[]>

Objects to which metadata is to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Map<PSObject>**

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name<String>**

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId<Guid>**

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
-----------	-------

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-MonitorService | % { Remove-MonitorServiceMetadata -Map $_.MetadataMap }  
Remove all metadata from all Service objects.
```

# Reset-MonitorDataStore

Jan 04, 2017

Refreshes the database string currently being used by the Monitor service.

## Syntax

```
Reset-MonitorDataStore [-DataStore] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the string for the database connection currently being used by the Monitor Service. Can only be called for secondary data stores.

There is no requirement for a database connection to be configured in order for this command to be used.

## Related topics

[Get-MonitorDataStore](#)

## Parameters

### **-DataStore**<String>

Specifies the database connection logical name to be used by the Monitor Service. Can be either be 'Site' or the logical name of the secondary data store. Specifying the site data store will display an error because this operation is not supported for site data stores.

Required?	true
Default Value	Site
Accept Pipeline Input?	false

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.Monitor.Sdk.ServiceStatus

The status of the specified data store.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

### DatabaseError

An error occurred in the service while attempting a database operation.

### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

### PermissionDenied

You do not have permission to execute this command.

### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

### CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Reset-MonitorDataStore -DataStore Secondary
```

```
OK
```

Refresh the database connection string for the Monitor Service.

# Reset-MonitorServiceGroupMembership

Jan 04, 2017

Reloads the access permissions and configuration service locations for the Monitor Service.

## Syntax

```
Reset-MonitorServiceGroupMembership [-ConfigServiceInstance] <ServiceInstance[]> [-LoggingId <Guid>] [-AdminAddress <String>] [-CommonParameters]
```

## Detailed Description

Enables you to reload Monitor Service access permissions and configuration service locations. The Reset-MonitorServiceGroupMembership command must be run on at least one instance of the service type (Monitor) after installation and registration with the configuration service. Without this operation, the Monitor services will be unable to communicate with other services in the XenDesktop deployment. When the command is run, the services are updated when additional services are added to the deployment, provided that the configuration service is not stopped. The Reset-MonitorServiceGroupMembership command can be run again to refresh this information if automatic updates do not occur when new services are added to the deployment. If more than one configuration service instance is passed to the command, the first instance that meets the expected service type requirements is used.

## Related topics

### Parameters

**-ConfigServiceInstance**<ServiceInstance[]>

Specifies the configuration service instance object that represents the service instance for the type 'InterService' that references a configuration service for the deployment.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.



Accept Pipeline Input?	false
------------------------	-------

## Input Type

Citrix.Monitor.Sdk.ServiceInstance[] Service instances containing a ServiceInstance object that refers to the central configuration service interservice interface can be piped to the Reset-MonitorServiceGroupMembership command.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoSuitableServiceInstance

None of the supplied service instance objects were suitable for resetting service group membership.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-MonitorServiceGroupMembership
```

Reset the service group membership for a service in a deployment where the configuration service is configured and running on the same machine as the service.

### ----- EXAMPLE 2 -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config -AdminAddress OtherServer.example.com | Reset-MonitorServiceGroupmembership
```

Reset the service group membership for a service in a deployment where the configuration service that is configured and running on a machine named 'OtherServer.example.com'.



# Set-MonitorConfiguration

Jan 04, 2017

Sets configuration settings that are used by the Monitor Service.

## Syntax

```
Set-MonitorConfiguration [-GroomSessionsRetentionDays <Int32>] [-GroomFailuresRetentionDays <Int32>] [-GroomLoadIndexesRetentionDays <Int32>] [-GroomDeletedRetentionDays <Int32>] [-GroomSummariesRetentionDays <Int32>] [-GroomMachineHotfixLogRetentionDays <Int32>] [-GroomMinuteRetentionDays <Int32>] [-DataCollectionEnabled <Boolean>] [-FullPollStartHour <Int32>] [-ResolutionPollTimeHours <Int32>] [-SyncPollTimeHours <Int32>] [-DetailedSqlOutputEnabled <Boolean>] [-CollectHotfixDataEnabled <Boolean>] [-GroomApplicationInstanceRetentionDays <Int32>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Sets the configuration settings used by the Monitor Service. Use these settings to modify the behavior of the service.

A database connection need not be configured for this command to be used.

### 注意

For parameters whose default values depend on the product edition, as noted here, the values of these parameters are set to the new default values when the edition is changed.

## Related topics

[Get-MonitorConfiguration](#)

## Parameters

### **-GroomSessionsRetentionDays**<Int32>

Determines how many days to keep Session and Connection records after the Session is terminated.

Required?	false
Default Value	7 for non-platinum, 90 for platinum.
Accept Pipeline Input?	false

### **-GroomFailuresRetentionDays**<Int32>

Determines how many days to keep MachineFailureLog and ConnectionFailureLog records after these are created.

Required?	false
-----------	-------

Default Value	7 for non-platinum, 90 for platinum.
Accept Pipeline Input?	false

**-GroomLoadIndexesRetentionDays<Int32>**

Determines how many days to keep LoadIndex records after these are created.

Required?	false
Default Value	7 for non-platinum, 90 for platinum.
Accept Pipeline Input?	false

**-GroomDeletedRetentionDays<Int32>**

Determines how many days to keep Machine, Catalog, DesktopGroup and Hypervisor entities around that have a LifecycleState of 'Deleted'. This also deletes any related Session, Connection, Summary, Failure or LoadIndex records.

Required?	false
Default Value	7 for non-platinum, 90 for platinum.
Accept Pipeline Input?	false

**-GroomSummariesRetentionDays<Int32>**

Determines how many days to keep DesktopGroupSummary, FailureLogSummary and LoadIndexSummary records at the daily granularity.

Required?	false
Default Value	7 for non-platinum, 90 for platinum.
Accept Pipeline Input?	false

**-GroomMachineHotfixLogRetentionDays<Int32>**

Determines how many days to keep Machine-Hotfix history records at the daily granularity.

Required?	false
-----------	-------

Default Value	90
Accept Pipeline Input?	false

**-GroomMinuteRetentionDays<Int32>**

Determines how many days to keep minute data.

Required?	false
Default Value	3
Accept Pipeline Input?	false

**-DataCollectionEnabled<Boolean>**

Starts / stops data collection. Stopping data collection turns off polling, and does not persist operational event data to the database.

Required?	false
Default Value	True
Accept Pipeline Input?	false

**-FullPollStartHour<Int32>**

Hour of day when Full Poll should begin.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ResolutionPollTimeHours<Int32>**

Start time for the Resolution Poll worker.

Required?	false
-----------	-------

Default Value	
Accept Pipeline Input?	false

**-SyncPollTimeHours<Int32>**

Start time for Sync Poll worker.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-DetailedSqlOutputEnabled<Boolean>**

Determines if the SqlLog should be enabled to send SQL statements to the CDF Trace

Required?	false
Default Value	False
Accept Pipeline Input?	false

**-CollectHotfixDataEnabled<Boolean>**

This setting determines if the hotfix inventory data should be collected and stored in the database or if it should be thrown away.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-GroomApplicationInstanceRetentionDays<Int32>**

Required?	false
Default Value	

Accept Pipeline Input?	false
------------------------	-------

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

Examples

----- **EXAMPLE 1** -----

C:\PS>Set-MonitorConfiguration -GroomSessionsRetentionDays 5 -GroomFailuresRetentionDays 4 ...  
 Updates the settings in the site database with the newly specified values.

# Set-MonitorDBConnection

Jan 04, 2017

Configures a database connection for the Monitor Service.

## Syntax

```
Set-MonitorDBConnection [-DBConnection] <String> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [[-DataStore] <String>] [<CommonParameters>]
```

## Detailed Description

Configures a connection to a database in which the Monitor Service can store its state. The service will attempt to connect and start using the database immediately after the connection is configured. The database connection string is updated to the specified value regardless of whether it is valid or not. Specifying an invalid connection string prevents a service from functioning until the error is corrected.

After a connection is configured, you cannot alter it without first clearing it (by setting the connection to \$null).

You do not need to configure a database connection to use this command.

## Related topics

[Get-MonitorServiceStatus](#)

[Get-MonitorDBConnection](#)

[Test-MonitorDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be used by the Monitor Service. Passing in \$null will clear any existing database connection configured.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Force**<SwitchParameter>

If present, allows the local administrator to set the connection string to null when there are problems contacting the database or other services.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.



Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**-DataStore<String>**

Specifies the logical name of the data store for the Monitor Service. Can be either be 'Site' or the logical name of the secondary data store.

Required?	false
Default Value	Site
Accept Pipeline Input?	false

**Return Values**

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Set-MonitorDBConnection command returns an object containing the status of the Monitor Service together with extra diagnostics information.

DBUnconfigured

The Monitor Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the Monitor Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Monitor Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the Monitor Service currently in use is incompatible with the version of the Monitor Service schema on the database. Upgrade the Monitor Service to a more recent version.

#### DBOlderVersionThanService

The version of the Monitor Service schema on the database is incompatible with the version of the Monitor Service currently in use. Upgrade the database schema to a more recent version.

#### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

#### OK

The Monitor Service is running and is connected to a database containing a valid schema.

#### Failed

The Monitor Service has failed.

#### Unknown

The status of the Monitor Service cannot be determined.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### InvalidDBConnectionString

The database connection string has an invalid format.

#### DatabaseConnectionDetailsAlreadyConfigured

There was already a database connection configured. After a configuration is set, it can only be set to \$null.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-MonitorDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Configures a database connection string for the Monitor Service.

----- **EXAMPLE 2** -----

```
c:\PS>Set-MonitorDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Configures an invalid database connection string for the Monitor Service.

# Set-MonitorServiceMetadata

Jan 04, 2017

Adds or updates metadata on the given Service.

## Syntax

```
Set-MonitorServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-MonitorServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-MonitorServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-MonitorServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Allows you to store additional custom data against given Service objects.

## Related topics

[Remove-MonitorServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<Service[]>

Objects to which metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name**<String>

Specifies the property name of the metadata to be added. The property must be unique for the Service specified. The property cannot contain any of the following characters \/:;#.\*?=<>|[]()"

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Value**<String>

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Map**<PSObject>

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with @{ "name1" = "val1"; "name2" = "val2" }) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]").

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

---

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

System.Collections.Generic.Dictionary[String,String]

Set-MonitorServiceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

**Notes**

If the command fails, the following errors can be returned.

**Error Codes**

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-MonitorServiceMetadata -ServiceHostId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Service with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Test-MonitorDBConnection

Jan 04, 2017

Tests a database connection for the Monitor Service.

## Syntax

```
Test-MonitorDBConnection [-DBConnection] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [[-DataStore] <String>] [<CommonParameters>]
```

## Detailed Description

Tests a connection to the database in which the Monitor Service can store its state. The service will attempt to connect to the database without affecting the current connection to the database.

You do not have to clear the connection to use this command.

## Related topics

[Get-MonitorServiceStatus](#)

[Get-MonitorDBConnection](#)

[Set-MonitorDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be tested by the Monitor Service.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.



Accept Pipeline Input?	false
------------------------	-------

**-DataStore<String>**

Specifies the logical name of the data store for the Monitor Service. Can be either be 'Site' or the logical name of the secondary data store.

Required?	false
Default Value	Site
Accept Pipeline Input?	false

**Return Values**

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Test-MonitorDBConnection command returns an object containing the status of the Monitor Service if the connection string of the specified data store were to be set to the string being tested, together with extra diagnostics information for the specified connection string.

DBRejectedConnection

The database rejected the logon attempt from the Monitor Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Monitor Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the Monitor Service currently in use is incompatible with the version of the Monitor Service schema on the database. Upgrade the Monitor Service to a more recent version.

DBOlderVersionThanService

The version of the Monitor Service schema on the database is incompatible with the version of the Monitor Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The Set-MonitorDBConnection command would succeed if it were executed with the supplied connection string.

Failed

The Monitor Service has failed.

Unknown

The status of the Monitor Service cannot be determined.

Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

##### InvalidDBConnectionString

The database connection string has an invalid format.

##### DatabaseError

An error occurred in the service while attempting a database operation.

##### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

##### PermissionDenied

You do not have permission to execute this command.

##### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

##### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

##### CommunicationError

There was a problem communicating with the remote service.

##### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

----- **EXAMPLE 1** -----

```
c:\PS>Test-MonitorDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Tests a database connection string for the Monitor Service.

----- **EXAMPLE 2** -----

```
c:\PS>Test-MonitorDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Tests an invalid database connection string for the Monitor Service.

# Citrix.Storefront.Admin.V1

Jan 04, 2017  
Overview

Name	Description
<a href="#">SfStorefrontSnapin</a>	This service short description
<a href="#">Sf Filtering</a>	Describes the common filtering options for XenDesktop cmdlets.

## Cmdlets

Name	Description
<a href="#">Add-SfServerToCluster</a>	Adds a new server to an existing cluster.
<a href="#">Add-SfStorefrontAddress</a>	Modifies a StoreFront address configuration by adding an additional StoreFront address to it.
<a href="#">Get-SfCluster</a>	Gets all Storefront clusters present in the site.
<a href="#">Get-SfDBConnection</a>	Gets the database string for the specified data store used by the Storefront Service.
<a href="#">Get-SfDBSchema</a>	Gets a script that creates the Storefront Service database schema for the specified data store.
<a href="#">Get-SfDBVersionChangeScript</a>	Gets a script that updates the Storefront Service database schema.
<a href="#">Get-SfInstalledDBVersion</a>	Gets a list of all available database schema versions for the Storefront Service.
<a href="#">Get-SfIsStorefrontInstalled</a>	Tells whether StoreFront Services and Privileged Service are installed.
<a href="#">Get-SfService</a>	Gets the service record entries for the Storefront Service.
<a href="#">Get-SfServiceAddedCapability</a>	Gets any added capabilities for the Storefront Service on the controller.
<a href="#">Get-SfServiceInstance</a>	Gets the service instance entries for the Storefront Service.
<a href="#">Get-SfServiceStatus</a>	Gets the current status of the Storefront Service on the controller.

Name	Description
<a href="#">Get-SfStorefrontAddress</a>	Gets the high-level description of a configuration for StoreFront addresses, based on a configuration byte array.
<a href="#">Get-SfTask</a>	Gets the task history for the Storefront Service.
<a href="#">New-SfCluster</a>	Creates new Storefront cluster with default set of services.
<a href="#">New-SfStorefrontAddress</a>	Creates a new StoreFront address configuration, specifying a single address.
<a href="#">Remove-SfServerFromCluster</a>	Removes server from the cluster.
<a href="#">Remove-SfServiceMetadata</a>	Removes metadata from the given Service.
<a href="#">Remove-SfTask</a>	Removes from the database completed tasks for the Storefront Service.
<a href="#">Remove-SfTaskMetadata</a>	Removes metadata from the given Task.
<a href="#">Reset-SfServiceGroupMembership</a>	Reloads the access permissions and configuration service locations for the Storefront Service.
<a href="#">Set-SfCluster</a>	Sets the parameters on the given cluster.
<a href="#">Set-SfDBConnection</a>	Configures a database connection for the Storefront Service.
<a href="#">Set-SfServiceMetadata</a>	Adds or updates metadata on the given Service.
<a href="#">Set-SfTaskMetadata</a>	Adds or updates metadata on the given Task.
<a href="#">Test-SfDBConnection</a>	Tests a database connection for the Storefront Service.

# about\_SfStorefrontSnapin

Jan 04, 2017

## TOPIC

about\_SfStorefrontSnapin

## SHORT DESCRIPTION

This service short description

## COMMAND PREFIX

All commands in this snap-in have the noun prefixed with 'Sf'.

## LONG DESCRIPTION

This service long description

# about\_Sf\_Filtering

Jan 04, 2017

## TOPIC

XenDesktop - Advanced Dataset Filtering

## SHORT DESCRIPTION

Describes the common filtering options for XenDesktop cmdlets.

## LONG DESCRIPTION

Some cmdlets operate on large quantities of data and, to reduce the overhead of sending all of that data over the network, many of the Get- cmdlets support server-side filtering of the results.

The conventional way of filtering results in PowerShell is to pipeline them into Where-Object, Select-Object, and Sort-Object, for example:

```
Get-<Noun> | Where { $_.Size = 'Small' } | Sort 'Date' | Select -First 10
```

However, for most XenDesktop cmdlets the data is stored remotely and it would be slow and inefficient to retrieve large amounts of data over the network and then discard most of it. Instead, many of the Get- cmdlets provide filtering parameters that allow results to be processed on the server, returning only the required results.

You can filter results by most object properties using parameters derived from the property name. You can also sort results or limit them to a specified number of records:

```
Get-<Noun> -Size 'Small' -SortBy 'Date' -MaxRecordCount 10
```

You can express more complex filter conditions using a syntax and set of operators very similar to those used by PowerShell expressions.

Those cmdlets that support filtering have the following common parameters:

-MaxRecordCount <int>

Specifies the maximum number of results to return.  
For example, to return only the first nine results use:

```
Get-<Noun> -MaxRecordCount 9
```

If not specified, only the first 250 records are returned, and if more are available, a warning is produced:

WARNING: Only first 250 records returned. Use -MaxRecordCount to retrieve more.

You can suppress this warning by using -WarningAction or by specifying a value for -MaxRecordCount.

To retrieve all records, specify a large number for -MaxRecordCount. As the value is an integer, you can use the following:

```
Get-<Noun> -MaxRecordCount [int]::MaxValue
```

-ReturnTotalRecordCount [<SwitchParameter>]

When specified, this causes the cmdlet to output an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. For example:

```
Get-<Noun> -MaxRecordCount 9 -ReturnTotalRecordCount
```

....

```
Get-<Noun> : Returned 9 of 10 items
```

```
At line:1 char:18
```

```
+ Get-<Noun> <<<< -MaxRecordCount 9 -ReturnTotalRecordCount
```

```
+ CategoryInfo          : OperationStopped: (:) [Get-<Noun>], PartialDataException
```

```
+ FullyQualifiedErrorId : PartialData,Citrix.<SDKName>.SDK.Get<Noun>
```

The count can be accessed using the TotalAvailableResultCount property:

```
$count = $error[0].TotalAvailableResultCount
```

-Skip <int>

Skips the specified number of records before returning results.

Also reduces the count returned by -ReturnTotalRecordCount.

-SortBy <string>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order, respectively. Ascending order is assumed if no prefix is present.

Sorting occurs before `-MaxRecordCount` and `-Skip` parameters are applied. For example, to sort by Name and then by Count (largest first) use:

```
-SortBy 'Name,-Count'
```

By default, sorting by an enumeration property uses the numeric value of the elements. You can specify a different sort order by qualifying the name with an ordered list of elements or their numeric values, or `<null>` to indicate the placement of null values.

Elements not mentioned are placed at the end in their numeric order. For example, to sort by two different enums and then by the object id:

```
-SortBy 'MyState(StateC,<null>,StateA,StateB),Another(0,3,2,1),Id'
```

`-Filter <String>`

This parameter lets you specify advanced filter expressions, and supports combination of conditions with `-and` and `-or`, and grouping with braces. For example:

```
Get-<Noun> -Filter 'Name -like "High*" -or (Priority -eq 1 -and Severity -ge 2)'
```

The syntax is close enough to PowerShell syntax that you can use script blocks in most cases. This can be easier to read as it reduces quoting:

```
Get-<Noun> -Filter { Count -ne $null }
```

The full `-Filter` syntax is provided below.

## EXAMPLES

Filtering by strings performs a case-insensitive wildcard match. Separate parameters are combined with an implicit `-and` operator. Normal PowerShell quoting rules apply, so you can use single or double quotes, and omit the quotes altogether for many strings. The order of parameters does not make any difference. The following are equivalent:

```
Get-<Noun> -Company Citrix -Product Xen*
Get-<Noun> -Company "citrix" -Product '[X]EN*'
Get-<Noun> -Product "Xen*" -Company "CITRIX"
Get-<Noun> -Filter { Company -eq 'Citrix' -and Product -like 'Xen*' }
```



See [about\\_Quoting\\_Rules](#) and [about\\_Wildcards](#) for details about PowerShell handling of quotes and wildcards.

To avoid wildcard matching or include quote characters, you can escape the wildcards using the normal PowerShell escape mechanisms (see [about\\_Escape\\_Characters](#)), or switch to a filter expression and the `-eq` operator:

```
Get-<Noun> -Company "Abc[*]"           # Matches Abc*
Get-<Noun> -Company "Abc`*"           # Matches Abc*
Get-<Noun> -Filter { Company -eq "Abc*" } # Matches Abc*
Get-<Noun> -Filter { Company -eq "A`"B`C" } # Matches A"B'C
```

Simple filtering by numbers, booleans, and TimeSpans perform direct equality comparisons, although if the value is nullable you can also search for null values. Here are some examples:

```
Get-<Noun> -Uid 123
Get-<Noun> -Enabled $true
Get-<Noun> -Duration 1:30:40
Get-<Noun> -NullableProperty $null
```

More comparisons are possible using advanced filtering with `-Filter`:

```
Get-<Noun> -Filter 'Capacity -ge 10gb'
Get-<Noun> -Filter 'Age -ge 20 -and Age -lt 40'
Get-<Noun> -Filter 'VolumeLevel -like "[123]"'
Get-<Noun> -Filter 'Enabled -ne $false'
Get-<Noun> -Filter 'NullableProperty -ne $null'
```

You can check boolean values without an explicit comparison operator, and you can also combine them with `-not`:

```
Get-<Noun> -Filter 'Enabled' # Equivalent to 'Enabled -eq $true'
Get-<Noun> -Filter '-not Enabled' # Equivalent to 'Enabled -eq $false'
```

See [about\\_Comparison\\_Operators](#) for an explanation of the operators, but note that only a subset of PowerShell operators are supported (`-eq`, `-ne`, `-gt`, `-ge`, `-lt`, `-le`, `-like`, `-notlike`, `-in`, `-notin`, `-contains`, `-notcontains`).

Enumeration values can either be specified using typed values or the string name of the enumeration value:

```
Get-<Noun> -Shape [Shapes]::Square
Get-<Noun> -Shape Circle
```

With filter expressions, typed values can be specified with simple variables or quoted strings. They also support enumerations with wildcards:

```
$s = [Shapes]::Square
Get-<Noun> -Filter { Shape -eq $s -or Shape -eq "Circle" }
Get-<Noun> -Filter { Shape -like 'C*' }
```

By their nature, floating point values, DateTime values, and TimeSpan values are best suited to relative comparisons rather than just equality. DateTime strings are converted using the locale and time zone of the user device, but you can use ISO8601 format strings (YYYY-MM-DDThh:mm:ss.sTZD) to avoid ambiguity. You can also use standard PowerShell syntax to create these values:

```
Get-<Noun> -Filter { StartTime -ge "2010-08-23T12:30:00.0Z" }
$d = [DateTime]"2010-08-23T12:30:00.0Z"
Get-<Noun> -Filter { StartTime -ge $d }
$d = (Get-Date).AddDays(-1)
Get-<Noun> -Filter { StartTime -ge $d }
```

Relative times are quite common and, when using filter expressions, you can also specify DateTime values using a relative format:

```
Get-<Noun> -Filter { StartTime -ge '-2' } # Two days ago
Get-<Noun> -Filter { StartTime -ge '-1:30' } # Hour and a half ago
Get-<Noun> -Filter { StartTime -ge '-0:0:30' } # 30 seconds ago
```

## ARRAY PROPERTIES

When filtering against list or array properties, simple parameters perform a case-insensitive wildcard match against each of the members. With filter expressions, you can use the -contains and -notcontains operators. Unlike PowerShell, these perform wildcard matching on strings.

Note that for array properties the naming convention is for the returned property to be plural, but the parameter used to search for any match is singular. The following are equivalent (assuming Users is an array property):

```
Get-<Noun> -User Fred*
Get-<Noun> -Filter { User -like "Fred*" }
Get-<Noun> -Filter { Users -contains "Fred*" }
```

You can also use the singular form with -Filter to search using other operators:

```
# Match if any user in the list is called "Frederick"
Get-<Noun> -Filter { User -eq "Frederick" }
# Match if any user in the list has a name alphabetically below 'F'
Get-<Noun> -Filter { User -lt 'F' }
```

## COMPLEX EXPRESSIONS

When matching against multiple values, you can use a sequence of comparisons joined with -or operators, or you can use -in and -notin:

```
Get-<Noun> -Filter { Shape -eq 'Circle' -or Shape -eq 'Square' }
$shapes = 'Circle','Square'
Get-<Noun> -Filter { Shape -in $shapes }
$sides = 1..4
Get-<Noun> -Filter { Sides -notin $sides }
```

Braces can be used to group complex expressions, and override the default left-to-right evaluation of -and and -or. You can also use -not to invert the sense of any sub-expression:

```
Get-<Noun> -Filter { Size -gt 4 -or (Color -eq 'Blue' -and Shape -eq 'Circle') }
Get-<Noun> -Filter { Sides -lt 5 -and -not (Color -eq 'Blue' -and Shape -eq 'Circle') }
```

## PAGING

The simplest way to page through data is to use the -Skip and -MaxRecordCount parameters. So, to read the first three pages of data with 10 records per page, use:

```
Get-<Noun> -Skip 0 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 10 -MaxRecordCount 10 <other filtering criteria>
Get-<Noun> -Skip 20 -MaxRecordCount 10 <other filtering criteria>
```

You must include the same filtering criteria on each call, and ensure that the data is sorted consistently.

The above approach is often acceptable, but as each call performs an independent query, data changes can result in records being skipped or appearing twice. One approach to improve this is to sort by a unique id field and then start the search for the next page at the unique id after the last unique id of the previous page. For example:

```
# Get the first page
Get-<Noun> -MaxRecordCount 10 -SortBy SerialNumber

SerialNumber ...
----- ---
A120004
A120007
... 7 other records ...
A120900

# Get the next page
Get-<Noun> -MaxRecordCount 10 -Filter { FirstName -gt 'A120900' }

SerialNumber ...
```

----- ---  
A120901  
B220000  
...

## FILTER SYNTAX DEFINITION

<Filter> ::= <ScriptBlock> | <ComponentList>

<ScriptBlock> ::= "{" <ComponentList> "}"

<ComponentList> ::= <Component> <AndOrOperator> <ComponentList> |

<Component>

<Component> ::= <NotOperator> <Factor> |

<Factor>

<Factor> ::= "(" <ComponentList> ")" |

<PropertyName> <ComparisonOperator> <Value> |  
<PropertyName>

<AndOrOperator> ::= "-and" | "-or"

<NotOperator> ::= "-not" | "!"

<ComparisonOperator>

::= "-eq" | "-ne" | "-le" | "-ge" | "-lt" | "-gt" |  
"-like" | "-notlike" | "-contains" | "-notcontains" |  
"-in" | "-notin"

<PropertyName> ::= <simple name of property>

<Value> ::= <string literal> | <numeric literal> |

<scalar variable> | <array variable> |  
"\$null" | "\$true" | "\$false"

Numeric literals support decimal and hexadecimal literals, with optional multiplier suffixes (kb, mb, gb, tb, pb).

Dates and times can be specified as string literals. The current culture determines what formats are accepted. To avoid any ambiguity, use strings formatted to the ISO8601 standard. If not specified, the current time zone is used.

Relative date-time string literals are also supported, using a minus sign followed by a TimeSpan. For example, "-1:30" means 1 hour and 30 minutes ago.

# Add-SfServerToCluster

Jan 04, 2017

Adds a new server to an existing cluster.

## Syntax

```
Add-SfServerToCluster -ClusterId <Guid> -ServerName <String> [-StorefrontUrl <Uri>] [-FarmName <String>] [-XmlServices <Uri[]>] [-RunAsynchronously <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Adds a new server to an existing cluster. Optionally updates Farm and Storefront Url. After operation succeeds, all servers are configured identically.

## Related topics

[Get-SfCluster](#)

[New-SfCluster](#)

[Remove-SfServerFromCluster](#)

[Set-SfCluster](#)

## Parameters

**-ClusterId**<Guid>

The id of the cluster to perform operation on.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-ServerName**<String>

The name of the server to join to existing cluster. The name must be one of the values returned by Get-SfCluster

Required?	true
Default Value	
Accept Pipeline Input?	false

**-StorefrontUri**<Uri>

The url that will be used by Receivers to contact Storefront. Http or https absolute urls are accepted.

Required?	false
Default Value	Server name and http binding.
Accept Pipeline Input?	false

**-FarmName**<String>

Name of the farm that will be used within Store service. Either both FarmName and XmlServices need to be specified or none of them.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-XmlServices**<Uri[]>

Collection of the url of xml services that will be used inside a farm. The urls need to be http or https, be absolute and share the same schema and port. Either both FarmName and XmlServices need to be specified or none of them.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-RunAsynchronously**<Boolean>

If set, the command will run asynchronously.

Required?	false
Default Value	false
Accept Pipeline Input?	false

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

Citrix.Storefront.Sdk.Task or Citrix.Storefront.DataModel.Cluster

Returns cluster description or a task, if ran asynchronously.

### Examples

#### ----- **EXAMPLE 1** -----

```
Add-SfServerToCluster -ClusterId (Guid) -ServerName NewSfServer -RunAsynchronously $true  
Adds "NewSfServer" to cluster with id (Guid).
```



# Add-SfStorefrontAddress

Jan 04, 2017

Modifies a StoreFront address configuration by adding an additional StoreFront address to it.

## Syntax

```
Add-SfStorefrontAddress [-ByteArray] <Byte[]> -Name <String> -Url <String> -Enabled <Boolean> -Description <String> [<CommonParameters>]
```

## Detailed Description

Use this command to transform an existing StoreFront configuration into a new configuration, where the new configuration contains one additional address. The original configuration is supplied as an input, along with the properties of the new StoreFront address being added. The cmdlet outputs the modified configuration, which can then be passed to the Citrix Broker Service using the Add-BrokerMachineConfiguration command.

This command does not, by itself, have any persistent effects within XenDesktop. To make the change persistent, the new configuration byte array must first be transformed into a machine configuration within the Citrix Broker Service. To do this, use the New-BrokerMachineConfiguration command. You can then use the Add-BrokerMachineConfiguration and Set-BrokerMachineConfiguration commands to fully associate the new configuration with a delivery group.

## Related topics

[New-SfStorefrontAddress](#)

[Get-SfStorefrontAddress](#)

[New-BrokerMachineConfiguration](#)

[Add-BrokerMachineConfiguration](#)

[Set-BrokerMachineConfiguration](#)

## Parameters

### **-ByteArray**<Byte[]>

Specifies the initial configuration, on which the new configuration is based. All of the addresses in the original configuration are also present in the new configuration, along with the additional address specified. This configuration byte array is obtained from an earlier call to New-SfStorefrontAddress, or from Get-BrokerMachineConfiguration.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

### **-Name**<String>

Specifies the name of the new StoreFront.

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-Url**<String>

Specifies the URL to the StoreFront, such as "https://mysite.com/Citrix/StoreWeb".

Required?	true
Default Value	
Accept Pipeline Input?	false

### **-Enabled**<Boolean>

Specifies if the new StoreFront address should be enabled for user access.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Description**<String>

Specifies a human-readable description of the new StoreFront.

Required?	true
Default Value	
Accept Pipeline Input?	false

**Input Type**

System.Byte[] This cmdlet accepts configurations as pipeline input, as an alternative to supplying the ByteArray parameter.

**Return Values**

System.Byte[]

This cmdlet outputs the new, modified configuration. This differs from the original configuration in that it contains the additional StoreFront address.

**Examples**

----- **EXAMPLE 1** -----

C:\PS> \$newConfiguration = Add-SfStorefrontAddress -ByteArray \$originalConfiguration -Url "https://mysite.com/Citrix/StoreWeb" -Description "This StoreFront delivers my corporate applications" -Name  
This command transforms the configuration byte array specified by \$originalConfiguration, adds the new StoreFront details, and stores the resulting configuration in \$newConfiguration.

# Get-SfCluster

Jan 04, 2017

Gets all Storefront clusters present in the site.

## Syntax

```
Get-SfCluster [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets all Storefront clusters present in the site. There is one special Cluster with null id that lists the servers that are not part to any cluster (they are available to join any).

## Related topics

[New-SfCluster](#)

[Add-SfServerToCluster](#)

[Remove-SfServerFromCluster](#)

[Set-SfCluster](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.Storefront.DataModel.Cluster[]

Returns array of clusters available in current deployment.

# Get-SfDBConnection

Jan 04, 2017

Gets the database string for the specified data store used by the Storefront Service.

## Syntax

```
Get-SfDBConnection [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the database connection string for the specified data store.

If the returned string is blank, no valid connection string has been specified. In this case the service is running, but is idle and awaiting specification of a valid connection string.

## Related topics

[Get-SfServiceStatus](#)

[Set-SfDBConnection](#)

[Test-SfDBConnection](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

system.string

The database connection string configured for the Storefront Service.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

NoDBConnections

The database connection string for the Storefront Service has not been specified.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-SfDBConnection
```

Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True  
Get the database connection string for the Storefront Service.

# Get-SfDBSchema

Jan 04, 2017

Gets a script that creates the Storefront Service database schema for the specified data store.

## Syntax

```
Get-SfDBSchema [-DatabaseName <String>] [-ServiceGroupName <String>] [-ScriptType <ScriptTypes>] [-LocalDatabase] [-Sid <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Gets SQL scripts that can be used to create a new Storefront Service database schema, add a new Storefront Service to an existing site, remove a Storefront Service from a site, or create a database server logon for a Storefront Service. If no Sid parameter is provided, the scripts obtained relate to the currently selected Storefront Service instance, otherwise the scripts relate to Storefront Service instance running on the machine identified by the Sid provided. When obtaining the Evict script, a Sid parameter must be supplied. The current service instance is that on the local machine, or that explicitly specified by the last usage of the -AdminAddress parameter to a Storefront SDK cmdlet. The service instance used to obtain the scripts does not need to be a member of a site or to have had its database connection configured. The database scripts support only Microsoft SQL Server, or SQL Server Express, and require Windows integrated authentication to be used. They can be run using SQL Server's SQLCMD utility, or by copying the script into an SQL Server Management Studio (SSMS) query window and executing the query. If using SSMS, the query must be executed in 'SMDCMD mode'. The ScriptType parameter determines which script is obtained. If ScriptType is not specified, or is FullDatabase, the script contains:

- o Creation of service schema
- o Creation of database server logon
- o Creation of database user
- o Addition of database user to Storefront Service roles

If ScriptType is Instance, the returned script contains:

- o Creation of database server logon
- o Creation of database user
- o Addition of database user to Storefront Service roles

If ScriptType is Evict, the returned script contains:

- o Removal of Storefront Service instance from database
- o Removal of database user

If ScriptType is Login, the returned script contains:

- o Creation of database server logon only

If the service uses two data stores they can exist in the same database. You do not need to configure a database before using this command.

## Related topics

[Set-SfDBConnection](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database for which the schema will be generated.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ServiceGroupName**<String>

Specifies the name of the service group to be used when creating the database schema. The service group is a collection of all the Storefront services that share the same database instance and are considered equivalent; that is, all the services within a service group can be used interchangeably.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ScriptType**<ScriptTypes>

Specifies the type of database script returned. Available script types are:

### Database

Returns a full database script that can be used to create a database schema for the Storefront Service in a database instance that does not already contain a schema for this service. The DatabaseName and ServiceGroupName parameters must be specified to create a script of this type.

### Instance

Returns a permissions script that can be used to add further Storefront services to an existing database instance that already contains the full Storefront service schema, associating the services to the Service Group. The Sid parameter can optionally be specified to create a script of this type.

### Login

Returns a database logon script that can be used to add the required logon accounts to an existing database instance that contains the Storefront Service schema. This is used primarily when creating a mirrored database environment. The

DatabaseName parameter must be specified to create a script of this type.

#### Evict

Returns a script that can be used to remove the specified Storefront Service from the database entirely. The DatabaseName and Sid parameters must be specified to create a script of this type.

Required?	false
Default Value	Database
Accept Pipeline Input?	false

#### -LocalDatabase<SwitchParameter>

Specifies whether the database script is to be used in a database instance run on the same controller as other services in the service group. Including this parameter ensures the script creates only the required permissions for local services to access the database schema for Storefront services.

Required?	false
Default Value	
Accept Pipeline Input?	false

#### -Sid<String>

Specifies the SID of the controller on which the Storefront Service instance to remove from the database is running.

Required?	false
Default Value	
Accept Pipeline Input?	true (ByValue)

#### -AdminAddress<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.



Accept Pipeline Input?	false
------------------------	-------

## Return Values

### Systemstring

A string containing the required SQL script for application to a database.

## Notes

The scripts returned support Microsoft SQL Server Express Edition, Microsoft SQL Server Standard Edition, and Microsoft SQL Server Enterprise Edition databases only, and are generated on the assumption that integrated authentication will be used.

If the ScriptType parameter is not included or set to 'FullDatabase', the full database script is returned, which will:

Create the database schema.

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist).

If the ScriptType parameter is set to 'Instance', the script will:

Create the user and the role (providing the schema does not already exist).

Create the logon (providing the schema does not already exist) and associate it with a user.

If the ScriptType parameter is set to 'Login', the script will:

Create the logon (providing the schema does not already exist) and associate it with a pre-existing user of the same name.

If the LocalDatabase parameter is included, the NetworkService account will be added to the list of accounts permitted to access the database. This is required only if the database is run on a controller.

If the command fails, the following errors can be returned.

## Error Codes

-----

### GetSchemasFailed

The database schema could not be found.

### ActiveDirectoryAccountResolutionFailed

The specified Active Directory account or Group could not be found.

### DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-SfDBSchema -DatabaseName MyDB -ServiceGroupName MyServiceGroup > c:\SfSchema.sql
```

Get the full database schema for site data store of the Storefront Service and copy it to a file called 'c:\SfSchema.sql'.

This script can then be used to create the schema in a pre-existing database named 'MyDB' that does not already contain a Storefront Service site schema.

### ----- EXAMPLE 2 -----

```
c:\PS>Get-SfDBSchema -DatabaseName MyDB -scriptType Login > c:\StorefrontLogins.sql
```

Get the logon scripts for the Storefront Service.

# Get-SfDBVersionChangeScript

Jan 04, 2017

Gets a script that updates the Storefront Service database schema.

## Syntax

```
Get-SfDBVersionChangeScript -DatabaseName <String> -TargetVersion <Version> [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Returns a database script that can be used to upgrade or downgrade the site or secondary schema for the Storefront Service from the current schema version to a different version.

## Related topics

[Get-SfInstalledDBVersion](#)

## Parameters

**-DatabaseName**<String>

Specifies the name of the database instance to which the update applies.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-TargetVersion**<Version>

Specifies the version of the database you want to update to.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

System.Management.Automation.PSObject

A PSObject containing the required SQL script for application to a database.

## Notes

The PSObject returned by this cmdlet contains the following properties:

- Script The raw text of the SQL script to apply the update, or null in the case when no upgrade path to the specified target version exists.
- NeedExclusiveAccess Indicates whether all services in the service group must be shut down during the update or not.
- CanUndo Indicates whether the generated script allows the updated schema to be reverted to the state prior to the update.

Scripts to update the schema version are stored in the database so any service in the service group can obtain these scripts. Extreme caution should be exercised when using update scripts. Citrix recommends backing up the database before attempting to upgrade the schema. Database update scripts may require exclusive use of the schema and so may not be able to execute while any Storefront services are running. However, this depends on the specific update being carried out.

After a schema update has been carried out, services that require the previous version of the schema may cease to operate. The ServiceState parameter reported by the Get-SfServiceStatus command provides information about service compatibility. For example, if the schema has been upgraded to a more recent version that a service cannot use, the service reports "DBNewerVersionThanService".

If the command fails, the following errors can be returned.

## Error Codes

-----

### NoOp

The operation was successful but had no effect.

### NoDBConnections

The database connection string for the Storefront Service has not been specified.

### DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
C:\PS> $update = Get-SfDBVersionChangeScript -DatabaseName MyDb -TargetVersion 1.0.75.0
```

```
C:\PS> $update.Script > update_75.sql
```

Gets an SQL update script to update the current schema to version 1.0.75.0. The resulting update\_75.sql script is suitable for direct use with the SQL Server SQLCMD utility.

# Get-SfInstalledDBVersion

Jan 04, 2017

Gets a list of all available database schema versions for the Storefront Service.

## Syntax

```
Get-SfInstalledDBVersion [-Upgrade] [-Downgrade] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns the current version of the Storefront Service database schema, if no flags are set, otherwise returns versions for which upgrade or downgrade scripts are available and have been stored in the database.

## Related topics

### Parameters

**-Upgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be updated should be returned.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-Downgrade**<SwitchParameter>

Specifies that only schema versions to which the current database version can be reverted should be returned.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.

Accept Pipeline Input?	false
------------------------	-------

## Return Values

### System.Version

The Get-SfInstalledDbVersion command returns objects containing the new definition of the Storefront Service database schema version.

Major <Integer>

Minor <Integer>

Build <Integer>

Revision <Integer>

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

Both the Upgrade and Downgrade flags were specified.

#### NoOp

The operation was successful but had no effect.

#### NoDBConnections

The database connection string for the Storefront Service has not been specified.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-SfInstalledDBVersion
```

```
Major Minor Build Revision
```

```
-----
```

```
5 6 0 0
```

Get the currently installed version of the Storefront Service database schema.

----- **EXAMPLE 2** -----

```
c:\PS>Get-SfInstalledDBVersion -Upgrade
```

```
Major Minor Build Revision
```

```
-----
```

```
6 0 0 0
```

Get the versions of the Storefront Service database schema for which upgrade scripts are supplied.



# Get-SflsStorefrontInstalled

Jan 04, 2017

Tells whether StoreFront Services and Privileged Service are installed.

## Syntax

```
Get-SflsStorefrontInstalled [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

## Related topics

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

bool

True if both StoreFront Services and Privileged Service are installed, false otherwise.

# Get-SfService

Jan 04, 2017

Gets the service record entries for the Storefront Service.

## Syntax

```
Get-SfService [-Metadata <String>] [-Property <String[]>] [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns instances of the Storefront Service that the service publishes. The service records contain account security identifier information that can be used to remove each service from the database.

A database connection for the service is required to use this command.

## Related topics

### Parameters

#### **-Metadata**<String>

Gets records with matching metadata entries.

The value being compared with is a concatenation of the key name, a colon, and the value. For example: `-Metadata "abc:x"` matches records with a metadata entry having a key name of "abc" and a value starting with the letter "x".

Required?	false
Default Value	
Accept Pipeline Input?	false

#### **-Property**<String[]>

Specifies the properties to be returned. This is similar to piping the output of the command through `Select-Object`, but the properties are filtered more efficiently at the server.

Required?	false
Default Value	
Accept Pipeline Input?	false

#### **-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See `about_Sf_Filtering` for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by `-ReturnTotalRecordCount`.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-Sort By**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

### **-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Sf\_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

Citrix.Storefront.Sdk.Service

The Get-SfServiceInstance command returns an object containing the following properties.

Uid <Integer>

Specifies the unique identifier for the service in the group. The unique identifier is an index number.

ServiceHostId <Guid>

Specifies the unique identifier for the service instance.

DNSName <String>

Specifies the domain name of the host on which the service runs.

MachineName <String>

Specifies the short name of the host on which the service runs.

CurrentState <Citrix.Fma.Sdk.ServiceCore.ServiceState>

Specifies whether the service is running, started but inactive, stopped, or failed.

LastStartTime <DateTime>

Specifies the date and time at which the service was last restarted.

LastActivityTime <DateTime>

Specifies the date and time at which the service was last stopped or restarted.

OSType

Specifies the operating system installed on the host on which the service runs.

OSVersion

Specifies the version of the operating system installed on the host on which the service runs.

ServiceVersion

Specifies the version number of the service instance. The version number is a string that reflects the full build version of the service.

DatabaseUserName <string>

Specifies for the service instance the Active Directory account name with permissions to access the database. This will be either the machine account or, if the database is running on a controller, the NetworkService account.

Sid <string>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

ActiveSiteServices <string[]>

Specifies the names of active site services currently running in the service. Site services are components that perform long-running background processing in some services. This field is empty for services that do not contain site services.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### PartialData

Only a subset of the available data was returned.

#### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

#### CouldNotQueryDatabase

The query required to get the database was not defined.

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-SfService
```

```
Uid          : 1
ServiceHostId : aef6f464-f1ee-4042-a523-66982e0cecd0
DNSName      : MyServer.company.com
MachineName  : MYSERVER
CurrentState  : On
LastStartTime : 04/04/2011 15:25:38
LastActivityTime : 04/04/2011 15:33:39
OSType       : Win32NT
OSVersion    : 6.1.7600.0
ServiceVersion : 5.1.0.0
DatabaseUserName : NT AUTHORITY\NETWORK SERVICE
SID          : S-1-5-21-2316621082-1546847349-2782505528-1165
ActiveSiteServices : {MySiteService1, MySiteService2...}
Get all the instances of the Storefront Service running in the current service group.
```

# Get-SfServiceAddedCapability

Jan 04, 2017

Gets any added capabilities for the Storefront Service on the controller.

## Syntax

```
Get-SfServiceAddedCapability [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables updates to the Storefront Service on the controller to be detected.

You do not need to configure a database connection before using this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

System.String

String containing added capabilities.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- **EXAMPLE 1** -----

```
c:\PS>Get-SfServiceAddedCapability
```

Get the added capabilities of the Storefront Service.



# Get-SfServiceInstance

Jan 04, 2017

Gets the service instance entries for the Storefront Service.

## Syntax

```
Get-SfServiceInstance [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns service interfaces published by the instance of the Storefront Service. Each instance of a service publishes multiple interfaces with distinct interface types, and each of these interfaces is represented as a ServiceInstance object. Service instances can be used to register the service with a central configuration service so that other services can use the functionality.

You do not need to configure a database connection to use this command.

## Related topics

### Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.Storefront.Sdk.ServiceInstance

The Get-SfServiceInstance command returns an object containing the following properties.

ServiceGroupUid <Guid>

Specifies the unique identifier for the service group of which the service is a member.

ServiceGroupName <String>

Specifies the name of the service group of which the service is a member.

ServiceInstanceUID <Guid>

Specifies the unique identifier for registered service instances, which are service instances held by and obtained from a

central configuration service. Unregistered service instances do not have unique identifiers.

ServiceType <String>

Specifies the service instance type. For this service, the service instance type is always Sf.

Address

Specifies the address of the service instance. The address can be used to access the service and, when registered in the central configuration service, can be used by other services to access the service.

Binding

Specifies the binding type that must be used to communicate with the service instance. In this release of XenDesktop, the binding type is always 'wcf\_HTTP\_kerb'. This indicates that the service provides a Windows Communication Foundation endpoint that uses HTTP binding with integrated authentication.

Version

Specifies the version of the service instance. The version number is used to ensure that the correct versions of the services are used for communications.

ServiceAccount <String>

Specifies the Active Directory account name for the machine on which the service instance is running. The account name is used to provide information about the permissions required for interservice communications.

ServiceAccountSid <String>

Specifies the Active Directory account security identifier for the machine on which the service instance is running.

InterfaceType <String>

Specifies the interface type. Each service can provide multiple service instances, each for a different purpose, and the interface defines the purpose. Available interfaces are:

SDK - for PowerShell operations

InterService - for operations between different services

Peer - for communications between services of the same type

Metadata <Citrix.Storefront.Sdk.Metadata[]>

The collection of metadata associated with registered service instances, which are service instances held by and obtained from a central configuration service. Metadata is not stored for unregistered service instances.

Notes

If the command fails, the following errors can be returned.

Error Codes

-----

## DatabaseError

An error occurred in the service while attempting a database operation.

## DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

## DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

## PermissionDenied

You do not have permission to execute this command.

## AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

## CommunicationError

There was a problem communicating with the remote service.

## ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

### ----- EXAMPLE 1 -----

```
c:\PS>Get-SfServiceInstance
```

```
Address      : http://MyServer.com:80/Citrix/StorefrontContract
Binding      : wcf_HTTP_kerb
InterfaceType : SDK
Metadata     :
MetadataMap  :
ServiceAccount : ENG\MyAccount$
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164
ServiceGroupName : MyServiceGroup
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000
ServiceType  : Sf
Version      : 1
```

```
Address      : http://MyServer.com:80/Citrix/StorefrontContract/IServiceApi
Binding      : wcf_HTTP_kerb
InterfaceType : InterService
Metadata     :
MetadataMap  :
```

ServiceAccount : ENG\MyAccount  
ServiceAccountSid : S-1-5-21-2406005612-3133289213-1653143164  
ServiceGroupName : MyServiceGroup  
ServiceGroupUid : a88d2f6b-c00a-4f76-9c38-e8330093b54d  
ServiceInstanceUid : 00000000-0000-0000-0000-000000000000  
ServiceType : Sf  
Version : 1

Get all instances of the Storefront Service running on the specified machine. For remote services, use the AdminAddress parameter to define the service for which the interfaces are required. If the AdminAddress parameter has not been specified for the runspace, service instances running on the local machine are returned.

# Get-SfServiceStatus

Jan 04, 2017

Gets the current status of the Storefront Service on the controller.

## Syntax

```
Get-SfServiceStatus [-AdminAddress <String>][<CommonParameters>]
```

## Detailed Description

Enables the status of the Storefront Service on the controller to be monitored. If the service has multiple data stores it will return the overall state as an aggregate of all the data store states. For example, if the site data store status is OK and the secondary data store status is DBUnconfigured then it will return DBUnconfigured.

## Related topics

[Set-SfDBConnection](#)

[Test-SfDBConnection](#)

[Get-SfDBConnection](#)

[Get-SfDBSchema](#)

## Parameters

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Get-SfServiceStatus command returns an object containing the status of the Storefront Service together with extra diagnostics information.

DBUnconfigured

The Storefront Service does not have a database connection configured.

DBRejectedConnection

The database rejected the logon attempt from the Storefront Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

#### InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Storefront Service schema has not been added to the database.

#### DBNotFound

The specified database could not be located with the configured connection string.

#### DBNewerVersionThanService

The version of the Storefront Service currently in use is incompatible with the version of the Storefront Service schema on the database. Upgrade the Storefront Service to a more recent version.

#### DBOlderVersionThanService

The version of the Storefront Service schema on the database is incompatible with the version of the Storefront Service currently in use. Upgrade the database schema to a more recent version.

#### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

#### OK

The Storefront Service is running and is connected to a database containing a valid schema.

#### Failed

The Storefront Service has failed.

#### Unknown

(0) The service status cannot be determined.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-SfServiceStatus
```

DBUnconfigured

Get the current status of the Storefront Service.

# Get-SfStorefrontAddress

Jan 04, 2017

Gets the high-level description of a configuration for StoreFront addresses, based on a configuration byte array.

## Syntax

```
Get-SfStorefrontAddress [-ByteArray] <Byte[]> [<CommonParameters>]
```

## Detailed Description

Use this command to convert a configuration byte array into a set of named property settings. The byte array will either have been retrieved from the Citrix Broker Service, or from the New-SfStorefrontAddress cmdlet.

## Related topics

[New-SfStorefrontAddress](#)

[Add-SfStorefrontAddress](#)

[New-BrokerMachineConfiguration](#)

[Add-BrokerMachineConfiguration](#)

[Set-BrokerMachineConfiguration](#)

## Parameters

**-ByteArray**<Byte[]>

Specifies the low-level byte array (blob) to be interpreted.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

## Input Type

System.Byte[] The cmdlet accepts the ByteArray parameter as pipeline input.

## Return Values

Citrix.Storefront.Sdk.SfStorefrontAddress

This cmdlet outputs one SfStorefrontAddress object for each address that is configured within the slot. Each object has the following properties:

Name - Specifies the name of the StoreFront.

Url - Specifies the URL to the StoreFront, such as "https://mysite.com/Citrix/StoreWeb".

Enabled - Specifies whether the StoreFront is enabled for users.

Description - Specifies the human-readable name of the StoreFront.

## Examples

----- **EXAMPLE 1** -----

```
C:\PS> $configuration = New-SfStorefrontAddress -Url "https://mysite.com/Citrix/StoreWeb" -Description "This StoreFront delivers my corporate applications" -Name "StoreFront1" -Enabled $true
```

```
C:\PS> Get-SfStorefrontAddress -ByteArray $configuration
```

```
Name      Url      Enabled Description
----      -
StoreFront1 https://mysite.com/Citrix/StoreWeb True This StoreFront delivers my corporate applications.
```

This example shows a new configuration byte array being created to specify a single StoreFront address. The configuration byte array is then provided as input to the Get-SfStorefrontAddress command, which interprets and outputs the same fields.



# Get-SfTask

Jan 04, 2017

Gets the task history for the Storefront Service.

## Syntax

```
Get-SfTask [-TaskId] <Guid> [-ReturnTotalRecordCount] [-MaxRecordCount <Int32>] [-Skip <Int32>] [-SortBy <String>] [-Filter <String>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Returns a list of tasks that have run or are currently running within the Storefront Service.

## Related topics

[Remove-SfTask](#)

[Add-SfTaskMetadata](#)

[Remove-SfTaskMetadata](#)

## Parameters

**-TaskId**<Guid>

Specifies the task identifier to be returned.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-ReturnTotalRecordCount**<SwitchParameter>

When specified, the cmdlet outputs an error record containing the number of records available. This error record is additional information and does not affect the objects written to the output pipeline. See [about\\_Sf\\_Filtering](#) for details.

Required?	false
Default Value	False
Accept Pipeline Input?	false

**-MaxRecordCount**<Int32>

Specifies the maximum number of records to return.

Required?	false
Default Value	250
Accept Pipeline Input?	false

**-Skip**<Int32>

Skips the specified number of records before returning results. Also reduces the count returned by -ReturnTotalRecordCount.

Required?	false
Default Value	0
Accept Pipeline Input?	false

**-SortBy**<String>

Sorts the results by the specified list of properties. The list is a set of property names separated by commas, semi-colons, or spaces. Optionally, prefix each name with a + or - to indicate ascending or descending order. Ascending order is assumed if no prefix is present.

Required?	false
Default Value	The default sort order is by name or unique identifier.
Accept Pipeline Input?	false

**-Filter**<String>

Gets records that match a PowerShell-style filter expression. See about\_Sf\_Filtering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

## **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### UnknownObject

One of the specified objects was not found.

#### PartialData

Only a subset of the available data was returned.

#### InvalidFilter

A filtering expression was supplied that could not be interpreted for this cmdlet.

#### CouldNotQueryDatabase

The query required to get the database was not defined.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration service.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

# New-SfCluster

Jan 04, 2017

Creates new Storefront cluster with default set of services.

## Syntax

```
New-SfCluster -ServerName <String> -FarmName <String> -XmlServices <Uri[]> [-StorefrontUrl <Uri>] [-RunAsynchronously <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

The server will have a default set of services containing fully-functionally Storefront server with Authentication, Store, Receiver for Web and Desktop Appliance.

## Related topics

[Get-SfCluster](#)

[Add-SfServerToCluster](#)

[Remove-SfServerFromCluster](#)

[Set-SfCluster](#)

## Parameters

**-ServerName**<String>

The name of the server to build a cluster on. The name must be one of the values returned by `Get-SfCluster`

Required?	true
Default Value	
Accept Pipeline Input?	false

**-FarmName**<String>

Name of the farm that will be used within Store service.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-XmlServices**<Uri[]>

Collection of the url of xml services that will be used inside a farm. The urls need to be http or https, be absolute and share the same schema and port.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-StorefrontUrl**<Uri>

The url that will be used by Receivers to contact Storefront. Http or https absolute urls are accepted.

Required?	false
Default Value	Server name and http binding.
Accept Pipeline Input?	false

**-RunAsynchronously**<Boolean>

If set, the command will run asynchronously.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

Citrix.Storefront.Sdk.Task or Citrix.Storefront.DataModel.Cluster

Returns cluster description or a task, if ran asynchronously.

### Examples

----- **EXAMPLE 1** -----

`New-SfCluster -FarmName "XdSiteName" -XmlServices http://farm1,http://farm2 -ServerName SfServer -RunAsynchronously $true`  
Creates a new cluster on server "SfServer". The server will have a default set of services containing fully-functionally Storefront server with Authentication, Store, Receiver for Web, Desktop Appliance site and the required infrastructure. The Store service will have a farm named "XdSiteName" that will contain servers farm1 and farm2, whose will be contacted using http on default port 80.

# New-SfStorefrontAddress

Jan 04, 2017

Creates a new StoreFront address configuration, specifying a single address.

## Syntax

```
New-SfStorefrontAddress -Name <String> -Url <String> -Enabled <Boolean> -Description <String> [<CommonParameters>]
```

## Detailed Description

Use this command when you want to create a new StoreFront configuration byte array from scratch, rather than modifying an existing one. You must define the URL for the StoreFront, and some additional details.

This command does not, by itself, have any persistent effects within XenDesktop. To make the change persistent, the new configuration byte array must first be transformed into a machine configuration within the Citrix Broker Service. To do this, use the `New-BrokerMachineConfiguration` command. You can then use the `Add-BrokerMachineConfiguration` and `Set-BrokerMachineConfiguration` commands to fully associate the new configuration with a delivery group.

## Related topics

[Add-SfStorefrontAddress](#)

[Get-SfStorefrontAddress](#)

[New-BrokerMachineConfiguration](#)

[Add-BrokerMachineConfiguration](#)

[Set-BrokerMachineConfiguration](#)

## Parameters

**-Name**<String>

Specifies the name of the new StoreFront.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Url**<String>

Specifies the URL to the StoreFront, such as "https://mysite.com/Citrix/StoreWeb".

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Enabled**<Boolean>

Specifies if the new StoreFront address should be enabled for user access.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Description**<String>

Specifies a human-readable description of the new StoreFront.

Required?	true
Default Value	
Accept Pipeline Input?	false



Input Type

None

Return Values

System.Byte[]

The new configuration set, with all of the given modifications applied.

Examples

----- EXAMPLE 1 -----

```
C:\PS> $configuration = New-SfStorefrontAddress -Url "https://mysite.com/Citrix/StoreWeb" -Description "This StoreFront delivers my corporate applications" -Name "StoreFront1" -Enabled $true
```

```
C:\PS> Get-SfStorefrontAddress -ByteArray $configuration
```

Name	Url	Enabled	Description
------	-----	---------	-------------

StoreFront1	https://mysite.com/Citrix/StoreWeb	True	This StoreFront delivers my corporate applications.
-------------	------------------------------------	------	-----------------------------------------------------

This example shows a new configuration byte array being created to specify a single StoreFront address. The configuration byte array is then provided as input to the Get-SfStorefrontAddress command, which interprets and outputs the same fields.

# Remove-SfServerFromCluster

Jan 04, 2017

Removes server from the cluster.

## Syntax

```
Remove-SfServerFromCluster -ClusterId <Guid> -ServerName <String> [-StorefrontUrl <Uri>] [-FarmName <String>] [-XmlServices <Uri[]>] [-RunAsynchronously <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Removes server from the cluster and propagates information to other servers. The configuration of the server is wiped out, so the server can be reused.

## Related topics

[Get-SfCluster](#)

[New-SfCluster](#)

[Add-SfServerToCluster](#)

[Set-SfCluster](#)

## Parameters

**-ClusterId**<Guid>

The id of the cluster to perform operation on.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-ServerName**<String>

The name of the server to remove from cluster. The name must be one of the values returned by Get-SfCluster.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-StorefrontUri**<Uri>

The url that will be used by Receivers to contact Storefront. Http or https absolute urls are accepted.

Required?	false
Default Value	Server name and http binding.
Accept Pipeline Input?	false

**-FarmName**<String>

Name of the farm that will be used within Store service. Either both FarmName and XmlServices need to be specified or none of them.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-XmlServices**<Uri[]>

Collection of the url of xml services that will be used inside a farm. The urls need to be http or https, be absolute and share the same schema and port. Either both FarmName and XmlServices need to be specified or none of them.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-RunAsynchronously**<Boolean>

If set, the command will run asynchronously.

Required?	false
Default Value	false
Accept Pipeline Input?	false

### **-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

### **-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Return Values

Citrix.Storefront.Sdk.Task or Citrix.Storefront.DataModel.Cluster

Returns cluster description or a task, if ran asynchronously.

### Examples

#### ----- **EXAMPLE 1** -----

```
Remove-SfServerFromCluster -ClusterId (Guid) -ServerName BrokenSfServer -RunAsynchronously $true
```

Removes Server "BrokenSfServer" from a Cluster with id (Guid).

# Remove-SfServiceMetadata

Jan 04, 2017

Removes metadata from the given Service.

## Syntax

```
Remove-SfServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfServiceMetadata [-InputObject] <Service[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Service.

## Related topics

[Set-SfServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<Service[]>

Objects to which metadata is to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Map<PSObject>**

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name<String>**

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId<Guid>**

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
-----------	-------

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-SfService | % { Remove-SfServiceMetadata -Map $_.MetadataMap }  
Remove all metadata from all Service objects.
```



# Remove-SfTask

Jan 04, 2017

Removes from the database completed tasks for the Storefront Service.

## Syntax

```
Remove-SfTask [-TaskId] <Guid> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Enables completed tasks that have run within the Storefront Service to be removed from the database.

## Related topics

[Get-SfTask](#)

[Add-SfTaskMetadata](#)

[Remove-SfTaskMetadata](#)

## Parameters

**-TaskId**<Guid>

Specifies the identifier for the task to be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByPropertyName)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

### Input Type

System.Management.Automation.PSObject Objects containing the TaskId parameter can be piped to the Remove-SfTask command.

### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### ServiceStatusInvalidDb

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration service.

#### OperationDeniedByConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

# Remove-SfTaskMetadata

Jan 04, 2017

Removes metadata from the given Task.

## Syntax

```
Remove-SfTaskMetadata [-TaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfTaskMetadata [-TaskId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfTaskMetadata [-InputObject] <Task[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfTaskMetadata [-InputObject] <Task[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability to remove metadata from the given Task.

## Related topics

[Set-SfTaskMetadata](#)

## Parameters

**-TaskId**<Guid>

Id of the Task

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<Task[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Map<PSObject>**

Specifies a dictionary of (name, value)-pairs for the properties. This can be either a hashtable (created with @{"name1" = "val1"; "name2" = "val2"}) or a string dictionary (created with new-object "System.Collections.Generic.Dictionary[String,String]"). The properties whose names match keys in the map will be removed.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name<String>**

The metadata property to remove.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId<Guid>**

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress<String>**

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
-----------	-------

Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### InvalidParameterCombination

The cmdlet parameters are inconsistent.

#### UnknownObject

One of the specified objects was not found.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-SfTask | % { Remove-SfTaskMetadata -Map $_.MetadataMap }  
Remove all metadata from all Task objects.
```

# Reset-SfServiceGroupMembership

Jan 04, 2017

Reloads the access permissions and configuration service locations for the Storefront Service.

## Syntax

```
Reset-SfServiceGroupMembership [-ConfigServiceInstance] <ServiceInstance[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Enables you to reload Storefront Service access permissions and configuration service locations. The Reset-SfServiceGroupMembership command must be run on at least one instance of the service type (Sf) after installation and registration with the configuration service. Without this operation, the Storefront services will be unable to communicate with other services in the XenDesktop deployment. When the command is run, the services are updated when additional services are added to the deployment, provided that the configuration service is not stopped. The Reset-SfServiceGroupMembership command can be run again to refresh this information if automatic updates do not occur when new services are added to the deployment. If more than one configuration service instance is passed to the command, the first instance that meets the expected service type requirements is used.

## Related topics

### Parameters

**-ConfigServiceInstance**<ServiceInstance[]>

Specifies the configuration service instance object that represents the service instance for the type 'InterService' that references a configuration service for the deployment.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
-----------	-------



Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Input Type

Citrix.Storefront.Sdk.ServiceInstance[] Service instances containing a ServiceInstance object that refers to the central configuration service interservice interface can be piped to the Reset-SfServiceGroupMembership command.

## Notes

If the command fails, the following errors can be returned.

### Error Codes

-----

#### NoSuitableServiceInstance

None of the supplied service instance objects were suitable for resetting service group membership.

#### DatabaseError

An error occurred in the service while attempting a database operation.

#### DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

## Examples

----- **EXAMPLE 1** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-SfServiceGroupMembership
```

Reset the service group membership for a service in a deployment where the configuration service is configured and running on the same machine as the service.

----- **EXAMPLE 2** -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config -AdminAddress OtherServer.example.com | Reset-SfServiceGroupmembership
```

Reset the service group membership for a service in a deployment where the configuration service that is configured and running on a machine named 'OtherServer.example.com'.

---

# Set-SfCluster

Jan 04, 2017

Sets the parameters on the given cluster.

## Syntax

```
Set-SfCluster -ClusterId <Guid> [-StorefrontUrl <Uri>] [-FarmName <String>] [-XmlServices <Uri[]>] [-RunAsynchronously <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Sets the parameters on the given cluster and propagate the changes to all servers within a given cluster.

## Related topics

[Get-SfCluster](#)

[New-SfCluster](#)

[Add-SfServerToCluster](#)

[Remove-SfServerFromCluster](#)

## Parameters

**-ClusterId**<Guid>

The id of the cluster to perform operation on.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-StorefrontUrl**<Uri>

The url that will be used by Receivers to contact Storefront. Http or https absolute urls are accepted.

Required?	false
Default Value	Server name and http binding.
Accept Pipeline Input?	false

**-FarmName**<String>

Name of the farm that will be used within Store service. Either both FarmName and XmlServices need to be specified or none of them.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-XmlServices**<Uri[]>

Collection of the url of xml services that will be used inside a farm. The urls need to be http or https, be absolute and share the same schema and port. Either both FarmName and XmlServices need to be specified or none of them.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-RunAsynchronously**<Boolean>

If set, the command will run asynchronously.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

Citrix.Storefront.Sdk.Task or Citrix.Storefront.DataModel.Cluster

Returns cluster description or a task, if ran asynchronously.

**Examples**

----- **EXAMPLE 1** -----

Set-SfCluster -StorefrontUrl http://SfUrl -ClusterId (Guid) -RunAsynchronously \$true

Sets a Storefront Url in a Cluster with id (Guid). Propagates changes to all servers that are part of the cluster.

# Set-SfDBConnection

Jan 04, 2017

Configures a database connection for the Storefront Service.

## Syntax

```
Set-SfDBConnection [-DBConnection] <String> [-Force] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Configures a connection to a database in which the Storefront Service can store its state. The service will attempt to connect and start using the database immediately after the connection is configured. The database connection string is updated to the specified value regardless of whether it is valid or not. Specifying an invalid connection string prevents a service from functioning until the error is corrected.

After a connection is configured, you cannot alter it without first clearing it (by setting the connection to \$null).

You do not need to configure a database connection to use this command.

## Related topics

[Get-SfServiceStatus](#)

[Get-SfDBConnection](#)

[Test-SfDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be used by the Storefront Service. Passing in \$null will clear any existing database connection configured.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Force**<SwitchParameter>

If present, allows the local administrator to set the connection string to null when there are problems contacting the database or other services.

Required?	false
Default Value	false
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level

operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Set-SfDBConnection command returns an object containing the status of the Storefront Service together with extra diagnostics information.

**DBUnconfigured**

The Storefront Service does not have a database connection configured.

**DBRejectedConnection**

The database rejected the logon attempt from the Storefront Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

**InvalidDBConfigured**

The expected stored procedures are missing from the database. This may be because the Storefront Service schema has not been added to the database.

**DBNotFound**

The specified database could not be located with the configured connection string.

**DBNewerVersionThanService**

The version of the Storefront Service currently in use is incompatible with the version of the Storefront Service schema on the database. Upgrade the Storefront Service to a more recent version.

**DBOlderVersionThanService**

The version of the Storefront Service schema on the database is incompatible with the version of the Storefront Service currently in use. Upgrade the database schema to a more recent version.

#### DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The Storefront Service is running and is connected to a database containing a valid schema.

Failed

The Storefront Service has failed.

Unknown

The status of the Storefront Service cannot be determined.

#### Notes

If the command fails, the following errors can be returned.

#### Error Codes

-----

#### InvalidDBConnectionString

The database connection string has an invalid format.

#### DatabaseConnectionDetailsAlreadyConfigured

There was already a database connection configured. After a configuration is set, it can only be set to \$null.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

----- **EXAMPLE 1** -----

```
c:\PS>Set-SfDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Configures a database connection string for the Storefront Service.

----- **EXAMPLE 2** -----

```
c:\PS>Set-SfDBConnection -DBConnection "Invalid Connection String"
```



Invalid database connection string format.

Configures an invalid database connection string for the Storefront Service.

# Set-SfServiceMetadata

Jan 04, 2017

Adds or updates metadata on the given Service.

## Syntax

```
Set-SfServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-SfServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-SfServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-SfServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

## Detailed Description

Allows you to store additional custom data against given Service objects.

## Related topics

[Remove-SfServiceMetadata](#)

## Parameters

**-ServiceHostId**<Guid>

Id of the Service

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<Service[]>

Objects to which metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Map<PSObject>**

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with `@{"name1" = "val1"; "name2" = "val2"}`) or a string dictionary (created with `new-object "System.Collections.Generic.Dictionary[String,String]"`).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name<String>**

Specifies the property name of the metadata to be added. The property must be unique for the Service specified. The property cannot contain any of the following characters `\;#.*?=<>|[]()`

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Value<String>**

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId<Guid>**

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the `Start-LogHighLevelOperation` and `Stop-LogHighLevelOperation` cmdlets.

Required?	false
-----------	-------

Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

System.Collections.Generic.Dictionary[String,String]

Set-SfServiceMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

**Notes**

If the command fails, the following errors can be returned.

**Error Codes**

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured

The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
c:\PS>Set-SfServiceMetadata -ServiceHostId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	-----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Service with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Set-SfTaskMetadata

Jan 04, 2017

Adds or updates metadata on the given Task.

## Syntax

```
Set-SfTaskMetadata [-TaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-SfTaskMetadata [-TaskId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-SfTaskMetadata [-InputObject] <Task[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-SfTaskMetadata [-InputObject] <Task[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

## Detailed Description

Provides the ability for additional custom data to be stored against given Task objects.

## Related topics

[Remove-SfTaskMetadata](#)

## Parameters

**-TaskId**<Guid>

Id of the Task

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

**-InputObject**<Task[]>

Objects to which the metadata is to be added.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Map<PSObject>**

Specifies a dictionary of (name, value)-pairs for the properties. This can either be a hashtable (created with `@{"name1" = "val1"; "name2" = "val2"}`) or a string dictionary (created with `new-object "System.Collections.Generic.Dictionary[String,String]"`).

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

**-Name<String>**

Specifies the property name of the metadata to be added. The property must be unique for the Task specified. The property cannot contain any of the following characters `\;#.*?=<>|[]()`

Required?	true
Default Value	
Accept Pipeline Input?	false

**-Value<String>**

Specifies the value for the property.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId<Guid>**

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the `Start-LogHighLevelOperation` and `Stop-LogHighLevelOperation` cmdlets.

Required?	false
-----------	-------

Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
Default Value	Localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

**Return Values**

System.Collections.Generic.Dictionary[String,String]

Set-SfTaskMetadata returns a dictionary containing the new (name, value)-pairs.

Key <string>

Specifies the name of the property.

Value <string>

Specifies the value for the property.

**Notes**

If the command fails, the following errors can be returned.

**Error Codes**

-----

InvalidParameterCombination

The cmdlet parameters are inconsistent.

UnknownObject

One of the specified objects was not found.

DatabaseError

An error occurred in the service while attempting a database operation.

DatabaseNotConfigured



The operation could not be completed because the database for the service is not configured.

#### DataStoreException

An error occurred in the service while attempting a database operation - communication with the database failed for various reasons.

#### PermissionDenied

You do not have permission to execute this command.

#### AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

#### ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

#### CommunicationError

There was a problem communicating with the remote service.

#### ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

#### Examples

##### ----- EXAMPLE 1 -----

```
c:\PS>Set-SfTaskMetadata -TaskId 4CECC26E-48E1-423F-A1F0-2A06DDD0805C -Name property -Value value
```

Key	Value
---	-----
property	value

Add metadata with a name of 'property' and a value of 'value' to the Task with the identifier '4CECC26E-48E1-423F-A1F0-2A06DDD0805C'.

# Test-SfDBConnection

Jan 04, 2017

Tests a database connection for the Storefront Service.

## Syntax

```
Test-SfDBConnection [-DBConnection] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

## Detailed Description

Tests a connection to the database in which the Storefront Service can store its state. The service will attempt to connect to the database without affecting the current connection to the database.

You do not have to clear the connection to use this command.

## Related topics

[Get-SfServiceStatus](#)

[Get-SfDBConnection](#)

[Set-SfDBConnection](#)

## Parameters

**-DBConnection**<String>

Specifies the database connection string to be tested by the Storefront Service.

Required?	true
Default Value	
Accept Pipeline Input?	false

**-LoggingId**<Guid>

Specifies the identifier of the high-level operation this cmdlet call forms a part of. Citrix Studio and Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of the Start-LogHighLevelOperation and Stop-LogHighLevelOperation cmdlets.

Required?	false
Default Value	
Accept Pipeline Input?	false

**-AdminAddress**<String>

Specifies the address of a XenDesktop controller the PowerShell snap-in will connect to. You can provide this as a host name or an IP address.

Required?	false
-----------	-------

Default Value	localhost. Once a value is provided by any cmdlet, this value becomes the default.
Accept Pipeline Input?	false

## Return Values

Citrix.Fma.Sdk.Utilities.Service.ServiceStatusInfo

The Test-SfDBConnection command returns an object containing the status of the Storefront Service if the connection string of the specified data store were to be set to the string being tested, together with extra diagnostics information for the specified connection string.

DBRejectedConnection

The database rejected the logon attempt from the Storefront Service. This may be because the service attempted to log on with invalid credentials or because a database has not been installed in the specified location.

InvalidDBConfigured

The expected stored procedures are missing from the database. This may be because the Storefront Service schema has not been added to the database.

DBNotFound

The specified database could not be located with the configured connection string.

DBNewerVersionThanService

The version of the Storefront Service currently in use is incompatible with the version of the Storefront Service schema on the database. Upgrade the Storefront Service to a more recent version.

DBOlderVersionThanService

The version of the Storefront Service schema on the database is incompatible with the version of the Storefront Service currently in use. Upgrade the database schema to a more recent version.

DBVersionChangeInProgress

A database schema upgrade is currently in progress.

OK

The Set-SfDBConnection command would succeed if it were executed with the supplied connection string.

Failed

The Storefront Service has failed.

Unknown

The status of the Storefront Service cannot be determined.

## Notes

If the command fails, the following errors can be returned.

## Error Codes

-----

InvalidDBConnectionString

The database connection string has an invalid format.

PermissionDenied

You do not have permission to execute this command.

AuthorizationError

There was a problem communicating with the Citrix Delegated Administration Service.

ConfigurationLoggingError

The operation could not be performed because of a configuration logging error.

CommunicationError

There was a problem communicating with the remote service.

ExceptionThrown

An unexpected error occurred. For more details, see the Windows event logs on the controller or the XenDesktop logs.

Examples

----- **EXAMPLE 1** -----

```
c:\PS>Test-SfDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Tests a database connection string for the Storefront Service.

----- **EXAMPLE 2** -----

```
c:\PS>Test-SfDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Tests an invalid database connection string for the Storefront Service.

# サードパーティ製品についての通知

Jan 04, 2017

XenAppおよびXenDesktop 7.6には、次のドキュメントで定義された条件の元でライセンスが有効になったサードパーティのソフトウェアが含まれている可能性があります。

- [XenApp 7.6 and XenDesktop 7.6 Third Party Notices](#)
- [FlexNet Publisher Documentation Supplement: Software Licenses](#)