

XenApp und XenDesktop 7.5

Apr 12, 2016

[Info zu diesem Release](#)

[Nicht in diesem Release enthaltene Features](#)

[Bekannte Probleme](#)

[XenApp 7.5 und frühere Versionen](#)

[Systemanforderungen](#)

[Grundlegende Konzepte](#)

[Planen](#)

[Datenbankfehlertoleranz](#)

[Zugriff auf Desktops und Anwendungen beim Fehlschlagen der Delivery Controllers](#)

[Security](#)

[Hosten von Desktops und Anwendungen](#)

[Active Directory](#)

[Drucken](#)

[StoreFront](#)

[Remote-PC-Zugriff](#)

[Referenzarchitekturen](#)

[Designanleitungen](#)

[Implementierungsanleitungen](#)

[Erstellen einer neuen Umgebung](#)

[Vorbereiten der Installation](#)

[Installieren von VDAs mit Skripts in Active Directory](#)

[Erstellen einer Site](#)

[Ändern des Speicherorts der sekundären Datenbanken](#)

[Installieren von VDA auf älteren Systemen](#)

[Entfernen von Komponenten](#)

[Upgrade](#)

[Upgrade von XenDesktop 5](#)

[Upgrade von XenDesktop 7](#)

Migration

[Details zum Datenimport und Datenexport](#)

[Exportieren aus einer XenDesktop 4-Farm](#)

[Bearbeiten der XML-Datei des Migrationstools](#)

[Importieren von XenDesktop 4-Daten](#)

[Aufgaben nach der Migration](#)

Desktop- und Anwendungsbereitstellung

[Einrichten von Maschinenkatalogen](#)

[Bereitstellungsgruppen](#)

[Gehostete Anwendungen](#)

[Verbessern der Benutzererfahrung für Mobilgeräte](#)

[Bereitstellen von Remote-PC-Zugriff für Benutzer](#)

[Verhindern des Benutzerzugriffs auf den lokalen Desktop](#)

[Server-VDI](#)

High Definition-Benutzererfahrung mit HDX

[HDX-Systemanforderungen](#)

[Optimieren der Grafik- und Multimediabereitstellung](#)

[Konfigurieren der Flash-Umleitung](#)

[HDX 3D Pro](#)

[OpenGL Software Accelerator](#)

[Optimieren von Audiofeatures](#)

[Zuweisen von Prioritäten für den Datenverkehr im Netzwerk](#)

[Konfigurieren des Dateizugriffs auf zugeordnete Clientlaufwerke](#)

Citrix Connector 7.5 für System Center Configuration Manager

[Info zu diesem Release](#)

[Systemanforderungen](#)

[Planen](#)

[Strategien und bewährte Methoden](#)

[Installieren von Citrix Connector](#)

[Upgrade](#)

Sichern und Wiederherstellen

Deinstallieren des Connectors

Erstellen von Anwendungen

Bereitstellen von Anwendungen für Maschinenkataloge

Veröffentlichen von Anwendungen

Überwachen und Problembehandlung

Lizenzierung

Verbindungen und Ressourcen

Verwalten von Maschinen

Verwalten von Desktopcomputerkonten

Aktualisieren eines Masterimages

Aktualisieren eines Maschinenkatalogs

Verwalten der Anwendungs- und Desktopbereitstellung

Einstellen der Energieverwaltung der Desktopbetriebssystemmaschinen

Importieren und Exportieren von Benutzerdaten

Verwalten von Maschinensitzungen

Aktivieren oder Deaktivieren des Wartungsmodus

Verwalten von Bereitstellungsgruppen-Ressourcen

Sichern und Einschränken des Zugriffs auf Maschinen einer Bereitstellungsgruppe

Herunterfahren und Neustart von Maschinen

Aktualisieren einer Bereitstellungsgruppe

Verwalten der Serverlast von Serverbetriebssystemmaschinen

Delivery Controller-Umgebung

Drucker

Bereitstellen von Druckern

Verwalten von Druckertreibern

Optimieren der Druckleistung

Anzeigen von Druckern und Verwalten der Druckwarteschlangen

Verwalten von Richtlinien

Navigieren durch die Citrix Richtlinien und Einstellungen

Verwalten von Citrix Richtlinienvorlagen

Erstellen von Richtlinien

Konfigurieren von Richtlinieneinstellungen

Vergleichen von Richtlinien und Vorlagen

Anwenden von Richtlinien

Ermitteln der auf eine Verbindung angewendeten Richtlinien

Verwalten von Richtlinien mit dem Gruppenrichtlinien-Editor

Standardrichtlinieneinstellungen

Referenztabelle

Richtlinieneinstellungen für Connector für Configuration Manager 2012

Einstellungen der Richtlinie "ICA"

Einstellungen der Richtlinie "Lastverwaltung"

Einstellungen der Richtlinie "Profilverwaltung"

Einstellungen der Richtlinie "Receiver"

Einstellungen der Richtlinie "Virtual Delivery Agent"

Konfigurieren von COM-Port- und LPT-Portumleitungseinstellungen in der Registrierung

Benutzerprofile

Erhalten der Sitzungsaktivität

Personal vDisk

Installation und Upgrade

Konfiguration und Verwaltung

Tools

Anzeigen, Meldungen und Problembehandlung

Konfigurationsprotokollierung

Delegierte Administration

Administratoren

Rollen

Geltungsbereiche

Berichte

IPv6

Konfigurieren der Zeitzoneneinstellungen

Konfigurieren von Verbindungstimern

Konfigurieren der USB-Unterstützung

Clientordnerumleitung

Sicheres Authentifizieren mit Smartcards

Systemanforderungen für Smartcards

Aktivieren der Smartcard-Verwendung

Überwachen und Problembehandlung

Überwachen von Umgebungen mit Director

Behandeln von Benutzerproblemen

Überwachen von Bereitstellungen

Überwachen von Personal vDisks

Verwenden der Überwachungsdienst-OData-API

Über die API verfügbare Daten

Zugriff auf Daten mit der API

Informationen über die Zeit- und Datumsbereiche

Schützen von Endpunkten mit SSL

Methoden

Bestimmen der Enumerationswerte

Beispiele

SDK

Erste Schritte mit dem SDK

Hilfe zu PowerShell-Cmdlets

Hinweise zu Drittanbietern

Citrix SCOM Management Pack für XenApp und XenDesktop

Citrix SCOM Management Pack für den Lizenzserver

Info zu XenApp 7.5 und XenDesktop 7.5

Sep 29, 2015

In diesem Release teilen sich XenApp und XenDesktop eine einheitliche Architektur und Verwaltung. XenApp ist eine Lösung zur Anwendungs- und Desktopvirtualisierung, mit der Windows-Anwendungen als sicherer mobiler Dienst auf beliebige Geräte in jedem Netzwerk geliefert werden. Die Bedienung per Toucheingabe wirkt bei solchen Anwendungen systemeigen. XenDesktop liefert Windows-Anwendungen und Desktops in Form eines sicheren mobilen Dienstes auf beliebige Geräte über jedes Netzwerk und vermeidet dabei die Kosten und Komplexitäten der konventionellen PC-Verwaltung.

Dieses Dokument enthält Informationen über die neuen Komponenten und Features in diesem Release.

Hinweis: Die Verfügbarkeit einiger Features hängt von Ihrer Produktedition und Ihren Lizenzen ab. Weitere Informationen zu den Features der einzelnen Editionen finden Sie auf [dieser Produktwebseite](#).

Lesen Sie die Systemanforderungen und die Liste der bekannten Probleme in diesem Dokument sorgfältig durch, bevor Sie mit der Planung beginnen.

Weitere Informationen über unterstützte Sprachen finden Sie unter [CTX119253](#).

Neu in XenApp 7.5

- **XenApp auf Basis von FlexCast Management Architecture:** Dieses Release von XenApp basiert auf der FlexCast Management Architecture (FMA), die in XenDesktop 7.5 verwendet wird. Daher bietet XenApp viele Features, die auch in XenDesktop verfügbar sind, z. B.:
 - Plattformunterstützung für die Anwendungsbereitstellung, Mobilität, Dienste, flexible Bereitstellung und Cloudverwaltung
 - Möglichkeit der Verwendung von Citrix HDX-Technologien für eine bessere Benutzererfahrung bei einer Vielzahl von Geräten und Netzwerken
 - Eine einzelne Verwaltungskonsole (Citrix Studio) zum Bereitstellen von Anwendungen und Desktops für die Cloud- und lokale Infrastruktur
 - Eine neu gestaltete Konsole zur Überwachung und Problembehandlung (Citrix Director) mit integrierten EdgeSight-Features

Informationen zu den Auswirkungen der Migration zur FMA bei diesem Release von XenApp finden Sie unter [Unterschiede zwischen XenApp 7.5 und früheren XenApp-Versionen](#).

Hinweis: Da sich die Architektur bei diesem Release von der früherer XenApp-Releases unterscheidet, können Sie kein Upgrade von früheren XenApp-Releases auf dieses Release durchführen.

- **Cloudbereitstellungen:** Dieses Release bietet die Möglichkeit der Hybridcloudbereitstellung unter Amazon Web Services (AWS) oder in einer beliebigen anderen mit CloudPlatform betriebenen öffentlichen oder privaten Cloud. Cloudbereitstellungen werden über die gleichen Konsolen konfiguriert, verwaltet und überwacht wie Bereitstellungen in einer herkömmlichen lokalen Virtualisierungsinfrastruktur.
- **Vollständige AppDNA-Unterstützung:** AppDNA ermöglicht die automatische Analyse von Anwendungen für Windows-Plattformen und eignet sich für die Anwendungsvirtualisierung über App-V, XenApp oder XenDesktop. Vollständige AppDNA-Funktionalität steht in bestimmten Editionen zur Verfügung.
- **Unterstützung für zusätzliche Virtualisierungsressourcen:** Es können Verbindungen mit VMware vSphere 5.5-Hypervisoren konfiguriert werden.
- **StoreFront 2.5:** Das Installationsmedium enthält die neueste Version von StoreFront.
- **Unterstützung für Webinterface 5.4:** Dieses Release unterstützt Webinterface 5.4. Webinterface wurden keine Erweiterungen oder zusätzliche Betriebssystemunterstützung hinzugefügt.

Neu in XenDesktop 7.5

- **Cloudbereitstellungen:** Dieses Release bietet die Möglichkeit der Hybridcloudbereitstellung unter Amazon Web Services (AWS) oder in einer beliebigen anderen mit CloudPlatform betriebenen öffentlichen oder privaten Cloud. Cloudbereitstellungen werden über die gleichen Konsolen konfiguriert, verwaltet und überwacht wie Bereitstellungen in einer herkömmlichen lokalen Virtualisierungsinfrastruktur.
- **Remote-Energiesteuerung für physische PCs:** Remote-PC-Zugriff unterstützt Wake-On-LAN, sodass physische PCs remote eingeschaltet werden können. Mit diesem optionalen Feature können Benutzer ihre Büro-PCs ausgeschaltet lassen, wenn sie nicht verwendet werden, um Energiekosten zu sparen. Außerdem ist ein Remotezugriff möglich, wenn Maschinen unabsichtlich ausgeschaltet wurden, z. B. bei witterungsbedingten Problemen. Dieses Feature wird auf PCs unterstützt, die Intel Active Management Technology (AMT) unterstützen, und auf solchen, auf denen Wake-On-LAN im BIOS aktiviert wurde. Microsoft System Center Configuration Manager 2012 bietet Zugriff zum Aufrufen von AMT-Energiebefehlen auf einem PC.
- **Vollständige AppDNA-Unterstützung:** AppDNA ermöglicht die automatische Analyse von Anwendungen für Windows-Plattformen und eignet sich für die Anwendungsvirtualisierung über App-V, XenApp oder XenDesktop. Vollständige AppDNA-Funktionalität steht in bestimmten Editionen zur Verfügung.
- **Unterstützung für zusätzliche Virtualisierungsressourcen:** Es können Verbindungen mit VMware vSphere 5.5-Hypervisors konfiguriert werden.
- **StoreFront 2.5:** Das Installationsmedium enthält die neueste Version von StoreFront.
- **Unterstützung für Webinterface 5.4:** Dieses Release unterstützt Webinterface 5.4. Webinterface wurden keine Erweiterungen oder zusätzliche Betriebssystemunterstützung hinzugefügt.

Die nachfolgenden Abschnitte und der Abschnitt [Nicht in diesem Release enthaltene Features](#) enthalten Informationen zu den Unterschieden zwischen XenDesktop 7.5, XenDesktop 7.1 sowie XenDesktop 7 und älteren XenDesktop-Releases.

Neu in XenDesktop 7.1

- **GPU:** Der Grafikprozessor (GPU) ermöglicht mit seinen umfassenden Grafikvirtualisierungsfunktionen eine überragende Erfahrung für Benutzer, die grafikintensive Anwendungen verwenden und häufig 3D-Modelle anpassen. Mit der GPU-Funktion können viele Benutzer von einer besseren Grafikdarstellung profitieren. Administratoren können Workflows auf die Fähigkeit zur Erstellung von GPU-fähigen Verbindungen hin prüfen und Maschinenkataloge bewerten, die mit einem über MSC bereitgestellten Masterimage erstellt wurden. Wenn Probleme auftreten, müssen Administratoren die Maschinen ggf. manuell erstellen.
- **vGPU:** Mit dem vGPU-Feature (virtueller Grafikprozessor) können mehrere virtuelle Maschinen die Grafikverarbeitungsleistung eines einzelnen physischen GPU direkt nutzen. Sie können den hardwarebeschleunigten vGPU-Zugriff für Windows-Desktop-VDI-Arbeitslasten verwenden. Diese echte gemeinsame Hardware-GPU-Nutzung bietet vollständige Windows 7- oder Windows 2008 R2 SP1-Desktops, die für Benutzer mit komplexen und anspruchsvollen Designanforderungen geeignet sind. Die gemeinsame GPU-Nutzung wird für NVIDIA GRID K1- und K2-Karten unterstützt und verwendet die gleichen NVIDIA-Grafiktreiber, die auf nicht-virtualisierten Betriebssystemen bereitgestellt werden.
- **Unterstützung für Windows Server 2012 R2 und Windows 8.1:** Stellen Sie eine High-Definition-Benutzererfahrung auf virtuellen Desktops und physischen Maschinen unter Windows 8.1 bereit. HDX 3D Pro wurde ebenfalls aktualisiert und unterstützt jetzt Windows 8.1.

Hinweis: Das vGPU-Feature wird nicht auf VMs unterstützt, die unter Windows Server 2012 R2 oder Windows 8.1 ausgeführt werden.

Neu in XenDesktop 7

- **Maschinenkatalogdesktops und -anwendungen**

- **Desktops:** Bereitstellen von verwalteten Desktops für mehrere, gleichzeitig verbundene Remotebenutzer. Serverbetriebssystemmaschinendesktops und Desktopbetriebssystemmaschinendesktops bieten Features wie z. B. SmoothRoaming, Sitzungszuverlässigkeit, Lizenzverbrauch, Authentifizierung an Receiver und Delivery Services sowie Desktop Viewer. Mit diesem Release werden Maschinenkataloge und Desktops für Windows-Serverbetriebssystemmaschinen als kostengünstige Virtualisierungslösung eingeführt, da sie eine große Anzahl von Remotebenutzern pro Server und Speicherressource zulassen. Dieses Release bietet auch Remote-PC-Desktops, mit denen Benutzer Remotezugriff auf Ressourcen auf ihren Büro-PCs von einem beliebigen Gerät mit Citrix Receiver haben.
- **Anwendungen:** Integrieren Sie die XenApp-Veröffentlichung in einer Bereitstellung. Stellen Sie mit diesem Feature mehreren, gleichzeitig verbundenen Remotebenutzern freigegebene gehostete Anwendungen bereit.

- **Verbesserungen in HDX**

- **Unterstützung für Windows Server 2012 und Windows 8:** Stellen Sie eine High-Definition-Benutzererfahrung auf virtuellen Desktops und physischen Maschinen unter Windows 8 bereit. HDX 3D Pro wurde ebenfalls aktualisiert und unterstützt jetzt Windows 8.
- **Desktopgestaltungsumleitung:** Dies ist eine Erweiterung für die Aero-Umleitung, die in XenDesktop 5.5 eingeführt wurde. Sie nutzt den Grafikprozessor (GPU) bzw. den integrierten Grafikprozessor (IGP) auf unterstützten Windows-Clientgeräten für die lokale DirectX-Grafikwiedergabe und stellt Benutzern eine nahtlosere Windows 7 Aero- bzw. Windows 8-Desktopdarstellung bereit. Dieses Feature stellt reichhaltige Multimedienanwendungen auf Benutzergeräten bereit und erhält gleichzeitig eine hohe Skalierbarkeit auf dem Server.
- **Clientseitiger Inhaltsabruf von Windows Media:** Ein Benutzergerät kann Multimediadateien direkt vom Quellenanbieter im Internet oder Intranet streamen, um die Netzwerkauslastung und Serverskalierbarkeit zu verbessern.
- **Multicastunterstützung:** Verringert den Bandbreitenverbrauch beim Streamen von Livevideos zu Geschäftsstellen. Die Multicastunterstützung ermöglicht die Übertragung einer Windows-Medienquelle an mehrere Benutzer.
- **Multimedia-Transcodierung in Echtzeit:** Ermöglicht das Streaming von Audio und Video zu Mobilgeräten und verbessert die Bereitstellung von Windows Media-Inhalt. Hostbasierte Transcodierung bietet eine nahtlose Benutzererfahrung, selbst bei extremen Netzwerkbedingungen. Zur Verbesserung der Serverskalierbarkeit erfolgt die Transcodierung im Grafikprozessor, wenn der Virtual Delivery Agent einen unterstützten Grafikprozessor für die Hardwarebeschleunigung hat.
- **Audio über UDP (User Datagram-Protokoll) für Serverbetriebssystemmaschinen:** Erweitert die Unterstützung für die Audioübermittlung über UDP/RTP an Serverbetriebssystemmaschinen. Dieses Feature bietet selbst in Umgebungen mit hohem Netzwerkaufkommen und Paketverlusten eine hervorragende Audioqualität für Echtzeitanwendungen wie Videokonferenzen und Streamingmedien.
- **Webcam-Videokomprimierung:** Verringert den Bandbreitenverbrauch und verbessert die Leistung bei Verwendung unterstützter Videokonferenzanwendungen.
- **HDX 3D Pro:** Stellt Anwendungen mit Grafikprozessoren für die Hardwarebeschleunigung auf dem Desktop bereit. Dies umfasst auch professionelle, auf OpenGL und DirectX basierende 3D-Grafikanwendungen.
- **Serverseitige Wiedergabe reichhaltiger Grafik- und Videodaten:** Bereitstellen reichhaltiger Grafikdaten (einschließlich Windows 8 oder Windows Aero) für virtuelle Desktopbenutzer auf jedem Client, jeder Plattform und bei jeder Bandbreite durch Wiedergabe des Inhalts auf dem Server. Verfügen Benutzer über die aktuelle Version von Citrix Receiver stellen sie auch eine verbesserte Leistung bei serverseitig wiedergegebenem Video fest.
- **Verbesserte Flash-Umleitung:** Die Umleitung von Adobe Flash-Inhalten zu Benutzergeräten für eine lokale Wiedergabe basierend auf dem Netzwerktyp und der Umgebung des Benutzers. Die Flash-Umleitung verringert die Server- und Netzwerklast, erhöht die Skalierbarkeit und gewährleistet gleichzeitig eine High-Definition-Benutzererfahrung.

- **Neues Installationsprogramm:** Ein optimiertes Installationsprogramm führt Sie durch die Installation der Kernkomponenten (Delivery Controller, Studio, Director, StoreFront und Lizenzserver) sowie von Virtual Delivery Agents.
- **Profilverwaltung:** Standardmäßig wird die Citrix Profilverwaltung 5.0 ohne Benutzereingriffe auf Masterimages installiert, wenn Sie Virtual Delivery Agent installieren. Die Profilverwaltung muss jedoch nicht als Profillösung verwendet werden.
- **Konfigurationsprotokollierung:** Erfassen von Site-Konfigurationsänderungen und Administratoraktivitäten in einer Konfigurationsprotokollierungsdatenbank. Sie können das Protokoll in Studio mit einer Reihe von Filtern anzeigen und HTML- und CSV-Berichte erstellen.
- **Director:** Überwachen Sie mit der neu gestalteten Director-Benutzeroberfläche mit integrierten EdgeSight-Features Bereitstellungen und beheben Sie Probleme.
 - Eine neue Ansicht "Helpdesk" bietet Helpdeskadministratoren eine verbesserte Benutzererfahrung bei der Problembehandlung, sodass Probleme mit Benutzern, Maschinen und Anwendungen unverzüglich gelöst werden können.
 - Volladministratoren haben Zugriff auf andere Ansichten, z. B. ein neu gestaltetes Dashboard mit einer grafischen Übersicht von Bereitstellungen an einer zentralen Stelle und die Seite "Trends", die mit Grafikdarstellungen eine verbesserte detaillierte Überwachung und Problembehandlung der gesamten Bereitstellung mit verschiedenen Zeitbereichen gestattet.
 - Die EdgeSight-Leistungsverwaltung bietet eine Verlaufsspeicherung und Trendberichte. Mit der Verlaufsspeicherung von Daten im Vergleich zur Echtzeitbewertung können Administratoren Trendberichte erstellen, u. a. Kapazitäts- und Integritätstrends.
 - Die EdgeSight-Netzwerkanalyse nutzt HDX Insight und stellt eine kontextbezogene Ansicht der Anwendungen und Desktops im Netzwerk bereit. Mit diesem Feature bietet Director eine erweiterte Analyse des ICA-Datenverkehrs in der XenDesktop-Bereitstellung.
- **Delegierte Administration:** Gruppieren von Objekten in Verwaltungsgeltungsbereiche. Dieses Feature bietet ein Verwaltungsmodell der Unternehmensklasse mit rollenbasierter Zugriffssteuerung, benutzerdefinierten Rollen mit konfigurierbaren Berechtigungen und differenzierter objektbasierter Steuerung.
- **Automatisches Update von Delivery Controller:** Virtual Delivery Agents werden automatisch benachrichtigt, wenn Controller der Site hinzugefügt oder aus ihr entfernt werden. Dieses Feature verhindert, dass der VDA von unbekannten Controllern gestartete Sitzungen ablehnt und dass aufgrund ungültiger Controllerinformationen VDA-Startverzögerungen oder -Fehler auftreten. Weitere Informationen zum Beibehalten der CNAME-Funktionalität finden Sie unter [CTX137960](#).
- **Clientordnerumleitung:** Ändern Sie die Zugriffsmethode für clientseitige Dateien in der hostseitigen Sitzung. Wenn Sie nur die Clientlaufwerkzuordnung auf dem Server aktivieren, werden clientseitige vollständige Volumes automatisch Hostlaufwerksbuchstaben zugeordnet. Wenn Sie die Clientordnerumleitung auf dem Server aktivieren und der Benutzer sie dann auf dem Benutzergerät konfiguriert, wird nur der von dem Benutzer festgelegte Teil des lokalen Volumes umgeleitet.
- **Verbesserte Zugriffssteuerungseinstellungen für Virtual Desktop:** Steuern Sie den Benutzerzugriff auf Serverbetriebssystemmaschinen und Desktopbetriebssystemmaschinen mit einer einfachen und einheitlichen Methode mit einer neuen optimierten Gruppe von Sicherheitseinstellungen.
- **Verbesserte und integrierte Fehlerberichterstattung:** Die Studio-Fehlerberichte stellen einen direkten Link zur Citrix Support-Website bereit. Wenn Benutzer einen Fehler feststellen, werden bei Auswahl der Option "Get Advice" Informationen zu dem Fehler an die Citrix Website gesendet. Die Informationen werden analysiert und der Benutzer wird auf eine Website mit Lösungsvorschlägen umgeleitet.
- **Verbesserungen in StoreFront**
 - **Desktopgerätesites:** Greifen Sie auf Desktopgerätesites über eine Website im StoreFront-Store zu. Die Website wird automatisch erstellt, wenn ein Store erstellt wird. Wenn beispielsweise ein Store den Pfad path/Citrix/Store hat, ist

der Pfad der Desktopgerätesite path/Citrix/StoreDesktopAppliance. Benutzer können ihre auf virtuellen Maschinen gehosteten Desktops auf Desktopgerätesites neu starten.

- **Database as a Service:** Benutzerabonnementdaten für jeden Store können auf den lokalen Datenträger des StoreFront-Servers mit dem neuen Abonnementstordienst geschrieben werden. Die Daten werden dann in der Servergruppe verteilt.
- **Unterstützung für IPv6:** Verbindung mit Clients und Kernkomponenten in IPv4- und IPv6-Umgebungen oder Umgebungen mit dualen Stapel (IPv4/IPv6).
- **Personal vDisk:** Personalisieren virtueller Desktops. Außerdem wird vor der Anmeldung ein dedizierter Speicherdatenträger erstellt, damit Benutzer ihre Daten, einschließlich Anwendungen, die sie selbst installieren, auf dem Desktop speichern können.
- **Unterstützung von Maschinenerstellungsdiensten (MCS) für die Aktivierung über Microsoft Key Management System (KMS):** Jede mit MCS erstellte virtuelle Maschine (VM) bietet eine eindeutige Aktivierung für das Windows-Betriebssystem und Office 2010, sodass das KMS-System jede VM als separate Maschine erfassen kann.
- **Unterstützung von in Citrix Mobility Pack konfigurierten Gruppenrichtlinien:** Informationen zu diesen Richtlinien für Citrix Receiver für Mobilgeräte finden Sie unter [Konfigurieren von Richtlinien für Mobilitätsfeatures](#).
- **Unterstützung für Mehrfingereingabe:** Unterstützung für Fingereingabefunktionen auf Multitouchcomputern einschließlich Microsoft Windows 7- und Windows 8-Arbeitsstationen und Windows Server 2012 für Virtual Delivery Agents.
- **Remote-PC-Zugriff:** Die automatisierte Verwaltung von Remote-PC-Zugriff ist vollständig in die Kernfunktionen von XenDesktop Delivery Controller und Studio integriert. Ein Remote-PC-Maschinenkatalog und eine Remote-PC-Bereitstellungsgruppe werden automatisch erstellt, wenn Sie eine Remote-PC-Bereitstellung einrichten. Sie können weitere Maschinenkataloge und Bereitstellungsgruppen später mit Studio hinzufügen. Dieser Prozess ersetzt die XML-Konfigurationsdatei und PowerShell-Skripts, die von Remote-PC-Zugriff in XenDesktop 5.6 FP1 verwendet wurden. Die aktuelle Version enthält außerdem eine Funktion für den Remotezugriff auf Büro-PCs mit Windows 8.
- **Unterstützung für Schnelle Benutzerumschaltung mit RDP-Verbindungen:** Dieses Microsoft Windows-Feature ermöglicht die gemeinsame Verwendung eines Desktops durch mehrere Benutzer, ohne dass Programme geschlossen oder Benutzer abgemeldet werden müssen. Administratoren können mit diesem Feature eine RDP-Verbindung mit dem VDA herstellen, um Probleme zu beheben und Updates zu installieren, ohne Tasks oder Programme, die von angemeldeten Benutzern verwendet werden, zu unterbrechen.
- **Kompatibilität mit der in Korea verwendeten AhnLab-Tastaturverschlüsselung:** Dieses Release ist mit der auf Virtual Desktop Infrastructure-Hosts (VDI) installierten Tastaturverschlüsselung kompatibel.
- **Universeller Druckserver:** Der Delivery Controller enthält jetzt die Funktion des universellen Druckservers. Sie müssen nur den universellen Druckserver auf den Druckserver installieren. Der universelle Druckserver unterstützt Windows Server 2012 und Windows 8.

Verwalten mehrerer XenApp- und XenDesktop-Versionen

Wenn die Umgebung neben der aktuellen Version frühere Versionen von XenApp oder XenDesktop enthält, kann die aktuelle Studio-Version nicht zum Konfigurieren und Verwalten der früheren Produktversionen verwendet werden. Gleichmaßen kann die aktuelle Bereitstellung nicht mit früheren Konsolen verwaltet und überwacht werden.

Sie können jedoch aktuelle Virtual Delivery Agents in Bereitstellungen mit früheren Versionen von XenDesktop-Controllern verwenden.

Mit dieser Director-Version können XenDesktop 5.x-VDA überwacht werden. Allerdings stehen mit den XenDesktop 5.x-VDA einigen Daten, u. a. die Anmeldedauer, nicht zur Verfügung.

Wenn Sie weiterhin frühere Versionen von XenApp oder XenDesktop ausführen, empfiehlt Citrix, dass Sie diese parallel mit

der aktuellen Site ausführen und die Verwaltungskonsolen für jedes Release für diese Site ausführen.

Verwenden Sie StoreFront zum Aggregieren von Anwendungen und Desktops aus den verschiedenen Produktversionen.

Bekannte und behobene Probleme

Siehe auch:

- [Bekannte Probleme](#)
- [Behobene Probleme](#)
- [Behobene Probleme \(Delivery Controller\)](#)
- [Behobene Probleme \(Virtual Delivery Agent\)](#)
- [Behobene Probleme \(Provisioning Services\)](#)
- [Behobene Probleme \(Receiver für Windows\)](#)
- [Behobene Probleme \(Profilverwaltung\)](#)

Nicht in diesem Release enthaltene Features

Sep 29, 2015

Die folgenden Features sind zurzeit nicht Teil dieses Releases oder werden nicht mehr unterstützt.

- **SecureICA-Verschlüsselung unter 128 Bit:** In früheren Releases war eine Verschlüsselung von Clientverbindungen für Basic-, 40-Bit-, 56-Bit- und 128-Bit-Verschlüsselung durch SecureICA möglich. Bei diesem Release steht die SecureICA-Verschlüsselung nur für die 128-Bit-Verschlüsselung zur Verfügung.
- **Direkte SSL-Verbindungen:** In früheren Releases konnten Administratoren SSL-Relay-Unterstützung für Verbindungen mit dem Webinterface und zwischen einem SSL-aktivierten Plug-In und jedem Server konfigurieren. Die folgenden Verbindungstypen werden in diesem Release nicht unterstützt:
- **Legacydrucken:** Die folgenden Druckfunktionen werden in diesem Release nicht unterstützt:
 - Abwärtskompatibilität für DOS-Clients und 16-Bit-Drucker, einschließlich Legacy-Clientdruckernamen.
 - Unterstützung für mit Windows 95 und Windows NT verbundene Drucker, einschließlich verbesserter erweiterter Druckereigenschaften und Win32FavorRetainedSetting.
 - Möglichkeit zum Aktivieren oder Deaktivieren automatisch gespeicherter und automatisch wiederhergestellter Drucker.
 - DefaultPrnFlag, eine Registrierungseinstellung für Server zum Aktivieren/Deaktivieren automatisch gespeicherter und automatisch wiederhergestellter Drucker, die in Benutzerprofilen auf dem Server gespeichert werden.
- **Secure Gateway:** In früheren Releases wurden mit Secure Gateway sichere Verbindungen zwischen dem Server und den Benutzergeräten hergestellt. Die Sicherung externer Verbindungen erfolgt nun mit NetScaler Gateway.
- **Spiegeln von Benutzern:** In früheren Releases steuerten Administratoren die Benutzer-zu-Benutzer-Spiegelung mit Richtlinien. In diesem Release ist das Spiegeln von Endbenutzern ein integriertes Feature in Director, wobei die Microsoft-Remoteunterstützung den Administratoren das Spiegeln und Beheben von Problemen auf nahtlos bereitgestellten Anwendungen und virtuellen Desktops gestattet.
- **Anonyme Benutzer:** In früheren Releases konnten Administratoren beim Veröffentlichen einer Anwendung Berechtigung für die Gruppe "Anonym" erteilen, mit der Gastbenutzer ohne Benutzerauthentifizierung auf Anwendungen zugreifen konnten. In diesem Release werden keine Gastberechtigungen unterstützt. Administratoren können Bereitstellungsgruppen über Studio konfigurieren und Bereitstellungsgruppen dann virtuelle Desktops und Anwendungen zuweisen.
- **PreLaunch-Sitzungen:** In früheren Releases konnte mit dem PreLaunch-Feature die Anwendungsstartzeit bei normalem oder hohem Netzwerkverkehr verkürzt werden. Dieses Feature ist in diesem Release nicht verfügbar.
- **Energie- und Kapazitätsverwaltung:** In früheren Releases wurde mit der Energie- und Kapazitätsverwaltung der Stromverbrauch reduziert und die Serverkapazität verwaltet. Microsoft Configuration Manager ist das Ersatztool für diese Funktion.
- **Flash v1-Umleitung:** Bei Clients, die nicht die Flash-Umleitung der zweiten Generation unterstützen (einschließlich Receiver für Windows vor Version 3.0, Receiver für Linux 11.100 und das Citrix Online-Plug-In 12.1) erfolgt ein Fallback auf serverseitige Wiedergabe für Legacyfeatures der Flash-Umleitung. VDAs in diesem Release unterstützen die Flash-Umleitungsfeatures der zweiten Generation.
- **Lokales Textecho:** Dieses Feature wurde zur Beschleunigung der Anzeige eingegebenen Texts auf Benutzergeräten bei Verbindungen mit hoher Latenz eingesetzt. Aufgrund von Verbesserungen am Grafiks subsystem und HDX SuperCodec ist dieses Feature in XenDesktop 7 nicht enthalten.
- **Virtuelle IP - Loopbackunterstützung:** In früheren Releases konnte mit dieser Richtlinieneinstellung für jede Sitzung eine eigene virtuelle Loopbackadresse für die Kommunikation zugelassen werden. Diese Richtlinie ist in diesem Release nicht verfügbar. Einen Workaround-Vorschlag für Windows Server 2012 finden Sie in folgendem Microsoft-Artikel: <http://social.technet.microsoft.com/wiki/contents/articles/15230.rds-ip-virtualization-in-windows-server-2012.aspx>.

- **SmartAuditor:** In früheren Releases konnte mit SmartAuditor die Bildschirmaktivität einer Benutzersitzung aufgezeichnet werden. Diese Komponente ist in diesem Release nicht verfügbar.
- **Single Sign-On:** Dieses Feature zur Gewährleistung der Kennwortsicherheit wird für Windows 8- und Windows Server 2012-Umgebungen nicht unterstützt. Es wird noch für Windows 2008 R2- und Windows 7-Umgebungen unterstützt, ist aber nicht in diesem Release enthalten. Es ist auf der Downloadwebsite von Citrix verfügbar:
<http://citrix.com/downloads>.
- **Oracle-Datenbankunterstützung:** Diese Version erfordert SQL-Datenbanken.
- **Systemüberwachung und -wiederherstellung (HMR):** In früheren Releases konnten von HMR Tests auf Servern einer Serverfarm durchgeführt werden, um deren Zustand zu überwachen und mögliche Integritätsrisiken zu ermitteln. In diesem Release bietet Director eine zentrale Ansicht der Systemintegrität, da die Überwachungs- und Warnungsfunktionen für die ganze Infrastruktur innerhalb der Director-Konsole dargestellt werden.
- **Benutzerdefinierte ICA-Dateien:** Benutzerdefinierte ICA-Dateien wurden verwendet, um direkte Verbindungen von Benutzergeräten (mit der ICA-Datei) zu einer bestimmten Maschine herzustellen. In diesem Release ist dieses Feature standardmäßig deaktiviert, kann aber für die normale Verwendung mit einer lokalen Gruppe aktiviert werden, oder im Modus für hohe Verfügbarkeit verwendet werden, falls der Controller nicht mehr verfügbar ist.
- **Management Pack für System Center Operations Manager (SCOM) 2007:** Das Management Pack für die Überwachung der Aktivität der Farmen mit SCOM unterstützt diesen Release nicht.
- **CNAME-Funktion:** Die Funktion CNAME wurde zuvor standardmäßig aktiviert. Bereitstellungen, die von CNAME-Einträgen für FQDN-Umleitung und von der Verwendung von NetBIOS-Namen abhängig sind schlagen möglicherweise fehl. Dieses Release bietet stattdessen die automatische Aktualisierung von Delivery Controllern, durch die die Liste der Delivery Controller automatisch aktualisiert wird und VDAs automatisch benachrichtigt werden, wenn Controller der Site hinzugefügt und aus ihr entfernt werden. Die automatischen Updates für Delivery Controller ist in den Citrix Richtlinien standardmäßig aktiviert, kann jedoch durch Erstellen einer Richtlinie deaktiviert werden.
Alternativ können Sie die CNAME-Funktion in der Registrierung wieder aktivieren, um mit der vorhandenen Bereitstellung fortzufahren und FQDN-Umleitung und die Verwendung von NetBIOS-Namen zuzulassen. Weitere Informationen finden Sie unter [CTX137960](#).
- **Assistent für schnelle Bereitstellung:** In früheren Releases von Studio konnte mit dieser Option eine schnelle Bereitstellung einer vollständig installierten XenDesktop-Bereitstellung erzielt werden. Durch den neuen vereinfachten Installations- und Konfigurationsworkflow in XenDesktop 7 ist der Assistent für schnelle Bereitstellung nicht mehr nötig.
- **Systemeigene Gerätetreiber auf Delivery Controllern:** Ermöglichen Upgrades ohne Neustart; es werden keine Treiber installiert.
- **Konfigurationsdatei für Remote PC-Dienst und PowerShell-Skript für die automatische Verwaltung:** Remote PC ist jetzt in Studio und den Delivery Controller integriert.
- **Workflow Studio:** In früheren Releases war Workflow Studio die grafischen Oberfläche für den Aufbau von Workflows für XenDesktop. Dieses Feature wird in diesem Release nicht unterstützt.

Nicht in der Citrix Lizenzierung 11.11.1 enthaltene Features

- **Citrix Lizenzkonfigurationsdienst:** Von diesem Dienst wurden Lizenzinformationen angezeigt. Er ermöglichte eine eingeschränkte Lizenzserververwaltung in Versionen von Desktop Studio vor diesem aktuellen Release von Studio. Der alte Dienst wird in diesem Release durch Citrix Webdienste für die Lizenzierung ersetzt, die ähnliche Funktionalität bieten. Das erweiterte Tool enthält auch zusätzliche Funktionen. Citrix empfiehlt ein Upgrade auf XenDesktop 7 oder den Einsatz der License Administration Console zum Verwalten von Lizenzen.

Nicht in Receiver enthaltenen Features

- **COM-Portzuordnung:** Mit der COM-Portzuordnung wurde der Zugriff auf COM-Ports auf dem Benutzergerät erlaubt

oder verhindert. Die COM-Portzuordnung wurde zuvor standardmäßig aktiviert. In diesem Release ist die Funktion standardmäßig deaktiviert. Einzelheiten finden Sie unter [Konfigurieren von COM-Port- und LPT-Portumleitungseinstellungen in der Registrierung](#).

- **LPT-Portzuordnung:** Mit der LPT-Portzuordnung wird der Zugriff von Legacyanwendungen auf LPT-Ports gesteuert. Die LPT-Portzuordnung wurde zuvor standardmäßig aktiviert. In diesem Release ist die LPT-Portzuordnung standardmäßig deaktiviert.
- **PCM-Audiocodec:** In diesem Release wird der PCM-Audiocodec nur von HTML5-Clients unterstützt.
- **Unterstützung für Microsoft ActiveSync**
- **Proxyunterstützung für ältere Versionen**, dies umfasst:
 - Microsoft Internet Security und Acceleration (ISA) 2006 (Windows Server 2003).
 - Oracle iPlanet-Proxyserver 4.0.14 (Windows Server 2003).
 - Squid-Proxyserver 3.1.14 (Ubuntu Linux Server 11.10).

Bekannte Probleme

Sep 29, 2015

Allgemeine Probleme

Die folgende Notiz gilt für alle Workarounds, bei denen ein Registrierungseintrag geändert werden muss:

Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab.

- Zum Konfigurieren eines nicht standardmäßigen HTTP/SOAP-Ports für den universellen Druckserverwebdienst müssen Sie mit PowerShell-Cmdlets die Sitzungsdruckerrichtlinie ändern.

Informationen zum Konfigurieren von Gruppenrichtlinien finden Sie im Abschnitt zur Verwendung des Group Policy SDKs in den

— Informationen zum SDK

. So stellen Sie den Richtlinienwert ein:

Set-ItemProperty LocalGpo:\Computer\Unfiltered\Settings\ICA\Printing\UniversalPrintServer\UpsHttpPort -name Wert -Value <Portnummer>. [#268593]

- Nachdem das Anhalten eines Hyper-V-Hosts aufgehoben wird, aktualisiert Microsoft System Center Virtual Machine Manager (VMM) möglicherweise nicht sofort den Gesamtzustand des Hosts. Dies kann sich auf die Verwendung der Maschinenerstellungsdienste (MCS) auswirken. Wenn ein Hyper-V-Host einen angehaltenen Zustand meldet, wird dieser Host nicht für das Bereitstellen von virtuellen Maschinen verwendet. Wenn alle Hosts angehalten sind, schlägt die Katalogerstellung fehl. Als Workaround können Sie den Stammclusterknoten oder den Hyper-V-Host-Knoten manuell aktualisieren. Auf dem Host ausgeführte Umgebungstests melden auch Hosts mit einem Zustand "angehalten". [#285696]
- Das Brokering gehosteter Anwendungen auf Desktopbetriebssystemmaschinen wird nicht mit dem Remotedesktopprotokoll (RDP) unterstützt. [#377108]
- Bei der Bereitstellung von virtuellen Desktop-VMs mit Provisioning Services verwendet der Setupassistent nur eine Speicherressource im Hostinginfrastrukturdatensatz, selbst wenn mehrere Speicherressourcen verfügbar sind. Wenn Sie Provisioning Services für die Bereitstellung verwenden, können Sie als Workaround Hostinginfrastrukturdatensätze mit nur einem Speicherdatensatz erstellen. [#400212]
- Dieses Release unterstützt nicht das Bereitstellen einer ISO-Datei, wenn die Clientlaufwerkzuordnung für Sitzungen unter Windows 8 konfiguriert ist. [#333111]
- Wenn der universelle Druckserver von Citrix aufgrund falscher Treiber fehlschlägt, versuchen Sie, diese Treiber unter der Druckertreiberisolation auszuführen. Weitere Informationen zur Konfiguration der Druckertreiberisolation finden Sie unter 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). [#381460]
- Die Richtlinieneinstellung für die Enhanced Desktop Experience hat keine Auswirkungen auf vorhandene Benutzer oder Administratorprofile. Löschen Sie als Workaround alle bereits vorhandenen Profile, bevor Sie die Einstellung aktivieren bzw. deaktivieren; löschen Sie auch die Profile, die für die VDA-Installation verwendet werden (nach der VDA-Installation). Wenn das integrierte Administratorkonto für die Installation von Virtual Delivery Agent (VDA) verwendet wurde, können Sie das Profil nicht löschen. In diesem Fall kann der Benutzer das Citrix Enhanced Desktop-Design nach der Anmeldung auswählen. [#363736]
- Wenn die Bildschirmauflösung betroffen ist, da die Auflösung der VDA-Desktopsitzung die Clientbildschirmauflösung übersteigt, ändern Sie die Auflösungseinstellung auf dem VDA-Desktop auf den unterstützten Höchstwert für den angeschlossenen Monitor. Sie können auch die Monitorkabel von der VDA-Grafikkarte abziehen. [#365877]
- Bildschirmschoner und der Energiesparmodus sind in Sitzungen deaktiviert. Bearbeiten Sie die Registrierung und erstellen Sie folgenden DWORD-Wert: HKLM\Software\Citrix\Graphics\SetDisplayRequiredMode = 0 Diese Änderung verhindert nicht, dass der Bildschirmschoner oder der Energiesparmodus auf dem Remotecomputer aktiviert wird. Wenn der Energiesparmodus aktiviert wird, wird die Remotesitzung erst aktualisiert, wenn der Benutzer eine Eingabe (Maus/Tastatur) macht, aber die Bildschirmangaben werden nicht entfernt. [#380550]
- Bei der versuchten Verwendung eines PowerShell SDK New-ProvScheme-Befehls oder eines MCS-Befehls von einem Remotecomputer, bevor die Hostadminadresse eingestellt ist, kann ein Fehler auftreten. Stellen Sie die Adminadresse mit dem Befehl Set-HypAdminConnection ein, bevor Sie den New-ProvScheme-Befehl ausführen. [#336902]
- In diesem Release wird die Verwendung dieses Produkts mit dem Microsoft RemoteFX vGPU-Feature in einem Hyper-V-Host nicht unterstützt. Greifen Sie stattdessen mit RDP, z. B. Hyper-V vGPU, auf die RemoteFX-Funktionalität zu. [#375577]
- Director meldet die Anzahl an GPU-Maschinen, die nicht gestartet werden konnten, auf der Dashboardseite im Bereich für Maschinenfehler. Diese Informationen werden jedoch nicht angezeigt, wenn Sie einen Drilldown ausführen, um Detailinformationen auf der Seite "Filter" anzeigen.

Sie können diese Informationen auch im historischen Trenddiagramm anzeigen. [#0434722]

- Wenn mehr GPU-Maschinen eines bestimmten GPU-Typs bereitgestellt oder zugewiesen werden als der XenServer-Ressourcenpool unterstützen kann, starten die GPU-Maschinen u. U. nicht. Benutzer, die einer Maschine zugewiesen wurden (einschließlich der, die Personal vDisk verwenden), die nicht gestartet werden konnte, können möglicherweise nicht über StoreFront auf ihre virtuellen Desktops zugreifen. [#0434505]
- Wenn Sie eine GPU-Maschine einschalten und sie startet nicht, wird möglicherweise die folgende Fehlermeldung angezeigt:

Exception 'Failure in PowerOn, PGPU_INSUFFICIENT_CAPACITY_FOR_VGPU

Dies bedeutet, dass unzureichende GPU-Ressourcen vorhanden sind, um eine weitere Maschine, die den GPU verwendet, zu starten. [#434509]

- Wenn sich mehrere Benutzer mit dem RDP- und ICA-Protokoll an einer Desktopbetriebssystemmaschine anmelden, und ein Benutzer die RDP-Sitzung sperrt, können sich Benutzer, die eine Verbindung mit ICA herstellen, nicht an der Sitzung anmelden. Benutzer sollten Benutzer die Verbindung mit RDP-Sitzungen trennen oder sich abmelden, um dieses Problem zu vermeiden. [#392311]
- Wenn Sie eine Anwendung mit einem VDA-Host unter Windows 8 veröffentlichen und die Anwendung Nicht-ASCII-Zeichen im Infobereich anzeigt, werden andere nicht in Verbindung stehende Zeichen nach der Wiederverbindung auch im Infobereich angezeigt. Melden Sie sich von der Sitzung ab und starten Sie die Anwendung neu, um dieses Problem zu vermeiden. [#387963]
- Das Deaktivieren der Sitzungs freigabe über einen Registrierungseintrag wird in diesem Release nicht unterstützt. Daher ist die Sitzungs freigabe immer aktiviert. Wenn der Registrierungsschlüssel auf das Deaktivieren der Sitzungs freigabe eingestellt ist, wird die erste Anwendung gestartet, aber nachfolgende Anwendungen werden nicht gestartet. Für dieses Problem gibt es keinen Workaround. [#383718]
- Wenn ein Benutzer sich mit einer getrennten Sitzung auf Windows Server 2012 wieder verbindet, besteht ein Arbeitsspeicherverlust im Desktop-Fenstermanager. Wenn sich Benutzer häufig mit einer längeren Sitzung verbinden und die Verbindung trennen, beispielsweise verwendet ein Benutzer dieselbe Sitzung im Büro und von zu Hause, können die Speicherressourcen auf dem Server verringert werden; dies führt zu langsamen Reaktionszeiten und einem möglichen Serverausfall. Dies ist ein Drittanbieterproblem im Microsoft Code. Weitere Informationen finden Sie unter <http://support.microsoft.com/KB/2855336> (Veröffentlichung im Juli 2013). [#374261]
- Wenn Sie Controllern einer Site hinzufügen oder von der Site entfernen, müssen alle StoreFront-Server im Cluster muss eingeschaltet sein. Sonst schlägt der Vorgang für den StoreFront-Teil fehl und Studio weist eine Ausnahme auf, die fälschlicherweise angibt, dass der Vorgang fehlgeschlagen ist. In solchen Fällen wird beim Studio-Neustart die Siteanzeige korrigiert. Wenn Sie einen StoreFront-Server nicht wieder online schalten können, entfernen Sie von der StoreFront-Konsole einer anderen Maschine im Cluster StoreFront vom fehlgeschlagenen Server. Zum Wiedereinbinden des fehlgeschlagenen Servers im Cluster installieren Sie StoreFront neu oder führen Sie die folgenden Befehle nacheinander aus: C:\Programme\Citrix\ReceiverStoreFront\Skrpts\ImportModules.ps1 und Clear-DSXDConfiguration. [#396193]
- In Bereitstellungen, in denen eine SQL Server 2012-Datenbank mit AlwaysOn für die hohe Verfügbarkeit konfiguriert ist, können Intervalle auftreten (während eines Datenbank-Failovers), in denen Sitevorgänge möglicherweise fehlschlagen, z. B. VDA-Energieverwaltung oder das Starten von Sitzungen. Wenn dieses Problem auftritt, wiederholen Sie den betroffenen Vorgang, wenn das Failover abgeschlossen ist. [#394787]
- Anwendungen schlagen ggf. auf Benutzergeräten fehl, wenn Sie die Profilstreaming-Richtlinie ohne eingestellte Antivirenausschlüsse für Microsoft Gruppenrichtliniendateien, wie z. B. NTUser.pol,

aktivieren. Als Workaround können Sie das Profilstreaming deaktivieren oder Antiviren-Ausschlussrichtlinien für diese Benutzer einstellen. Weitere Informationen finden Sie unter <http://support.microsoft.com/kb/822158>. Lesen Sie insbesondere den Abschnitt "Deaktivieren des Scannens von mit Gruppenrichtlinien verknüpften Dateien", um weitere Informationen zum Ausschluss der Dateien NTUser.pol und Registry.pol zu erhalten. [#399152]

- Beim Arbeiten über Windows PowerShell 3.0 funktioniert die Hilfeinfo nicht. Dies ist ein Drittanbieterproblem von Microsoft. Andere Hilfethemen sind nicht betroffen. [#408866]
- Wenn Sie ein Upgrade von XenDesktop 5 auf dieses Release (oder auf XenDesktop 7.1 oder XenDesktop 7.0) durchführen, führt eine gehostete Anwendung, die sowohl einer freigegebenen als auch einer privaten Bereitstellungsgruppe zugewiesen ist, zu einem Bereitstellungsgruppenkompatibilitätsfehler während des Upgrades. Sie vermeiden diesen Fehler, indem Sie gehostete Anwendungen entweder einer freigegebenen oder einer privaten Bereitstellungsgruppe (aber nicht beiden) zuweisen. [# 419424]
- Windows Store-Apps, die vom Store auf dem virtuellen Desktop installiert wurden, starten nicht, wenn der virtuelle Desktop neu gestartet oder heruntergefahren und neu gestartet wird. Alle Apps, die nach dem Neustart des virtuellen Desktops installiert wurden, werden ebenfalls nicht gestartet. [# 413003]
- Kacheln des Startmenüs werden möglicherweise nicht angezeigt, wenn Sie sich bei einer Windows 8-Maschine anmelden und sich vorher bereits bei einer Windows 8.1-Maschine mit der gleichen Bitanzahl angemeldet und abgemeldet hatten. Sie verhindern dieses Problem, indem Sie es vermeiden, Windows 8.1-Profile auf Windows 8-Maschinen zu verwenden. [#422043]
- Wenn die Profilverwaltung aktiviert ist, werden Anmeldeskripts für Sitzungen, in denen Windows Server 2012 R2 oder Windows 8.1 ausgeführt wird, standardmäßig um fünf Minuten verzögert. Wenn die Sitzung verfügbar ist, ist der Schritt mit der Anmeldeskriptdauer in Director nicht verfügbar. Die Verzögerung wird von der Richtlinie zum Konfigurieren der Anmeldeskriptverzögerung gesteuert (Aktiviert = 0). [#407978]
- Desktops können u. U. nicht auf Computern starten, auf denen Windows 8.1 ausgeführt wird und Microsoft Softwareupdateverwaltung installiert ist. Sie vermeiden dieses Problem, indem Sie in Windows 8.1 im Eigenschaftendialogfeld für Taskleiste und Navigation auf der Registerkarte **Navigation** alle Optionen im Bereich der **Startseite** auswählen. [#408439]
- Wenn Sie bei dem Versuch, über das Remotedesktopprotokoll eine Verbindung mit einer vermittelten Sitzung herzustellen, die Fehlermeldung "Sie können auf diese Sitzung nicht zugreifen, da keine Lizenzen vorhanden sind" erhalten, deaktivieren Sie die folgenden Einstellungen in C:\inetpub\wwwroot\Citrix\Store\App_Data\default.ica: [#422212 und #403855]:
 - RDPConnection=false
 - RDP-RedirectDrives=false
 - RDP-RedirectDynamicDrives=false
- Der Wert für die Richtlinie "Schwellenwert für persistenten Cache" ist in Studio fälschlicherweise mit KBit/s beschriftet. Der richtige Wert ist "Bits pro Sekunde" (Bit/s). [# 429478]
- Wenn Sie nach dem Upgrade von XenDesktop 7.0 auf eine neuere Version im Studio-Bereich "Häufige Aufgaben" auf "Site testen" klicken, werden Fehler zum Schema der Überwachungsdienstdatenbank angezeigt. Sie können diese Fehler ignorieren, da sie keinen Einfluss auf die Site haben. Stellen Sie die Datenbank nicht aus dem aktuellen Backup wieder her, wenn der Bericht ausschließlich die folgenden Fehler enthält.

Objektdefinition für "StoredProcedure MonitorData.DeleteMachinesByCatalogId" stimmt nicht mit dem erwarteten Referenzschema überein.

Stellen Sie die Datenbank \ aus dem aktuellen Backup wieder her.

Objektdefinition für "StoredProcedure MonitorData.DeregisterMachinesByCatalogId" stimmt nicht mit dem erwarteten Referenzschema überein.

Stellen Sie die Datenbank \ aus dem aktuellen Backup wieder her.

Objektdefinition für "StoredProcedure MonitorData.RemoveMachinesByDesktopGroupId" stimmt nicht mit dem erwarteten Referenzschema überein.

Stellen Sie die Datenbank \ aus dem aktuellen Backup wieder her.

Objektdefinition für "StoredProcedure MonitorData.TerminateSessionsBySid" stimmt nicht mit dem erwarteten Referenzschema überein.

Stellen Sie die Datenbank \ aus dem aktuellen Backup wieder her. [#429471]

- Die standardmäßige Virtual Private Cloud (VPC) muss vorhanden sein, damit Maschinenerstellungsdienste (MCS) Maschinen bereitstellen kann. Wenn Sie die Standard-VPC aus AWS löschen, funktioniert MCS nicht mehr und der Support für AWS muss verständigt werden, um die Standard-VPC wiederherzustellen. [#437085]
- Bei der Auswahl eines benutzerdefinierten Datumsbereichs für einen Konfigurationsprotokollierungsbericht wird möglicherweise ein falscher Kalender angezeigt. [#452399]
- Verwenden Sie bei der Bereitstellung von StoreFront keine Unicode-Zeichen in einem Servernamen. [#452972]
- Unter Amazon Web Services oder CloudPlatform bereitgestellte Maschinen werden nicht angehalten, wenn die Energieverwaltungseinstellung ihrer Bereitstellungsgruppe für die Trennung der Verbindung auf "Anhalten" festgelegt ist. [#453780]
- Wenn Sie ein Upgrade von VDAs der Version 7 oder 7.1, die auf physischen Maschinen installiert sind (einschließlich für Remote-PC-Zugriff verwendete Maschinen), auf Version 7.5 durchführen, werden möglicherweise der Machine Identity Service Agent und Personal vDisk-Treiber installiert. Dies kann dazu führen, dass die VDAs eine Sicherheitsprüfung nicht bestehen. Sie verhindern dieses Problem, indem Sie das Upgrade der VDAs über die Befehlszeilenschnittstelle unter Verwendung der Option /EXCLUDE "Personal vDisk", "Machine Identity Service" durchführen. Siehe [Manuelles Upgrade von VDAs auf physischen Maschinen](#). [#455472, #455553]
- Verbindungszeiten (Timer für längste Verbindung, Verbindungstimer und Trennungstimer) funktionieren möglicherweise auf Windows Server 2012-Maschinen mit VDAs für Windows-Serverbetriebssysteme nicht und verursachen unerwartetes Sitzungstimeoutverhalten. [#471698]
- Benutzerverbindungen mit VDAs unter Windows Server 2012 R2 können fehlschlagen, wenn die Citrix Richtlinie **Maximal zugelassene Farbtiefe** aktiviert ist. Die Richtlinie **Maximal zugelassene Farbtiefe** gilt nicht für VDAs mit einem Windows Display Driver Model-Treiber für die primäre Anzeige, z. B. VDAs, auf denen Windows Server 2012 R2 ausgeführt wird.

Installationsprobleme

- Beim Installieren von Virtual Delivery Agent für Serverbetriebssysteme tritt möglicherweise Fehler 1935 auf. Grund: Abwärtskompatibilitätsfehler in Microsoft Visual C++ 2005 Redistributable. Weitere Informationen finden Sie auf der Microsoft Website oder versuchen Sie das Problem mit Windows Update zu beheben. [354833]
- Nach einer erfolgreichen Installation von VDA für Windows-Serverbetriebssysteme (aber vor dem Neustart der Maschine) enthält das Windows-Ereignisprotokoll möglicherweise mehrere Fehlermeldungen (z. B. TermService 1035 oder 1036, die angeben, dass der Terminalserver-Listenerstack ausgefallen war oder das Erstellen der Sitzung fehlgeschlagen ist. Wenn keine anderen Installationsfehler enthalten sind, können Sie diese Meldungen im Ereignisprotokoll ignorieren. [#374134]
- Wenn Sie ein Upgrade von XenDesktop 5.x durchführen, stellen Sie sicher, dass Desktop Studio von XenDesktop 5.x geschlossen ist, bevor Sie das Upgrade durchführen. Sonst wird Studio beim Upgrade ggf. unerwartet geschlossen. [#389374]
- Bei der Installation von Virtual Desktop Agent-Version 5.6 auf einem 64-Bit-System unter Windows Vista oder Windows XP sollten Sie Citrix Receiver nicht in die Installation einschließen, da die Installation ggf. nicht abgeschlossen wird. Führen Sie diese Schritte aus:
 1. Installieren Sie den Virtual Desktop Agent, deaktivieren Sie jedoch das Kontrollkästchen "Citrix Receiver" im Installationsassistenten.
 2. Installieren Sie Receiver (CitrixReceiverEnterprise.exe) vom Ordner "Citrix Receiver und Plug-Ins > Windows > Receiver" auf dem Produktinstallationsmedium.
 3. Wenn Sie das Offline-Plug-In benötigen, installieren Sie CitrixOfflinePlugin.exe vom Ordner "Citrix Receiver und Plug-Ins > Windows > Offline Plug-In" auf dem Produktinstallationsmedium. Wenn Sie als Installationspeicherort nicht den Standard (C:\Programme (x86)\Citrix) angeben, muss das Zielverzeichnis vor dem Start der Installation erstellt werden.Wenn Sie aus Versehen eine VDA-Installation mit aktiviertem Kontrollkästchen "Citrix Receiver" starten, beenden Sie die ausgeführte Anwendung mit Windows Task-Manager. [#381625]
- Beim Upgrade wird ein Administrator, der in XenDesktop 5.6 deaktiviert wurde, möglicherweise ohne Rolle oder Geltungsbereich in die neuere Version verschoben. Überprüfen Sie die angezeigten Administratoren in Studio und bearbeiten Sie sie ggf. [#394765]
- Wenn Sie ein Upgrade auf einem Controller durchführen, schließen Sie Studio und starten es neu, damit das System erkennt, dass der Controller aktualisiert wurde. Registrieren Sie den aktualisierten Controller, indem Sie den Controller auswählen und dann auf "Controller registrieren" klicken. [#382898]
- Wenn Sie die Profilverwaltung aktivieren und Benutzer feststellen, dass die Windows 8-Standardanwendungen (z. B. Wetter, Nachrichten und Bing) bei der ersten Anmeldung aber nicht bei

nachfolgenden Anmeldungen starten, müssen Sie dieses Feature neu konfigurieren. Dieses Problem wurde in Umgebungen beobachtet, in denen das Benutzerprofil nicht permanent ist und in denen der Ordner AppData\Local nicht ausgeschlossen wurde (Standardeinstellung). Fügen Sie als Workaround die Ordner AppData-Ordner\Local\Packages und AppData\Local\Microsoft\Application Shortcuts als Ausschlüsse hinzu. [#394802]

- Bei der Installation von Virtual Desktop Agent oder bei einem Upgrade auf Windows 7 werden Sie möglicherweise in einem Dialogfeld aufgefordert, den Computer neu starten, um die Änderungen zu übernehmen. Klicken Sie auf "Später neu starten", um das Upgrade fortzusetzen; wählen Sie nicht "Jetzt neu starten". [#396553]
- Das Installieren von VDAs über eine Active Directory-Gruppenrichtlinie mit einzelnen MSIs wird nicht empfohlen und kann fehlschlagen. Citrix empfiehlt die Verwendung der Startskripts, die auf dem Produktinstallationsmedium bereitgestellt werden (siehe [Installieren oder Entfernen von Virtual Delivery Agents in Active Directory mit Skripten](#)). [#383432, #372136]
- Der Abschluss der Optimierungsphase während der VDA-Installation kann lange dauern. In einigen Tests hat dies ungefähr 30 Minuten gedauert und kann länger dauern. Dies tritt auf, wenn der VDA auf einem Image mit dem Windows-Betriebssystem installiert wird, oder wenn im Installationsassistenten auf der Seite Features die Option Leistung optimieren ausgewählt wurde. Dies führt dazu, dass der Microsoft Native Image Generator (Ngen) ausgeführt wird. Wenn dieses Problem auftritt, warten Sie auf den Abschluss der Installation. Citrix empfiehlt, dass Sie Ngen auf dem VDA-Basisimage vor der Bereitstellung virtueller Desktops ausführen, um Verzögerungen zu vermeiden, die bei der Hintergrundausführung des Programms auf bereitgestellten virtuellen Desktops auftreten. [#381437]
- Das Erstellen eines Maschinenkatalogs schlägt bei Verwendung eines ODX-aktivierten Storage Area Network in einer Umgebung fehl, in der VMM 2012 SP1 und Hyper-V 2012 ausgeführt werden. Folgen Sie den Anweisungen in <http://support.citrix.com/article/CTX139333>, um dieses Problem zu beheben. [#424040]
- Wenn während eines Upgrades eine Fehlermeldung zu einem nicht gefundenen PICAlsPorticaV2-Einstiegspunkt angezeigt wird, können Sie sie bedenkenlos ignorieren. Schließen Sie das Upgrade ab und starten Sie den Computer neu, wenn Sie dazu aufgefordert werden. [#423947]
- Der VDA für Windows-Desktopbetriebssysteme wird möglicherweise unter Evaluierungsversionen von Windows 8 nicht installiert. Auf dem Bildschirm mit den Installationsoptionen wird die Meldung "Kann auf diesem Betriebssystem nicht installiert werden" angezeigt. Dieses Problem tritt auf, weil das Installationsprogramm die Windows 8-Evaluierungsversion fälschlicherweise als nicht unterstütztes Betriebssystem identifiziert. Ein Hotfix für dieses Problem ist im Knowledge Center-Artikel CTX139660 verfügbar. Mit dem Hotfix installieren Sie einen Patch für das Installationsprogramm, damit Sie es auf Windows-Evaluierungsversionen ausführen können.
- Während der Produktinstallation können bei von Provisioning Services erstellten Maschinen Fehler auftreten, wenn .NET Framework 3.5 nicht vor der Installation vorhanden war. Zum Umgehen dieses Problems stellen Sie sicher, dass die .NET Framework-Versionen 2.0, 3.0, 3.5, 4.0, 4.5 und 4.5.1 installiert sind. [#442639, #447851]

Probleme bei Remote-PC-Zugriff

- Nach dem Upgrade von XenDesktop 5.6 FP1 wird der Name des Remote-PC-Zugriff-Dienstadministrators möglicherweise nicht richtig angezeigt. Dies wirkt sich nicht auf den Betrieb aus. [#437948]
- Wenn eine Büromaschine in den Ruhezustand versetzt wurde, kann anschließend der Start einer Remote-PC-Zugriff-Sitzung fehlschlagen. Als Lösungsansatz bei Desktopmaschinen, auf denen eine Fehlermeldung angezeigt wird, versuchen Sie noch einmal, die Sitzung zu starten. Starten Sie Laptops, auf denen dauerhaft ein grauer Neuverbindungsbildschirm angezeigt wird, erneut (entweder remote über die Administratorkonsole durch Erzwingen des Herunterfahrens und eines Neustarts oder lokal mit dem Netzschalter). Dies kann zu Datenverlust führen. [#441154]
- Bei Einsatz des Aktivierungsproxys anstelle von Intel Active Management Technology (AMT) oder Wake-On-LAN-Paketen wird eine Maschine möglicherweise nicht reaktiviert. Dies ist ein von Microsoft System Center Configuration Manager verursachtes Problem. [#441412]
- Wenn die Energieverwaltung für Remote PC-Kataloge aktiviert ist, können Maschinen, die sich in einem anderen Subnetz als der Controller befinden, ggf. nicht per subnetzgesteuertes Broadcast gestartet werden. Wenn Sie eine subnetzübergreifende Energieverwaltung mit subnetzgesteuertem Broadcast benötigen und AMT nicht unterstützt wird, versuchen Sie es mit dem Aktivierungsproxy oder Unicast. [#453820]

Probleme mit Studio

- Der Studio-Start ist verzögert, wenn die Maschine, auf der Studio installiert ist, keine Internetverbindung hat. (Aus Sicherheitsgründen signiert Citrix .NET-basierte Komponenten mit einer Authenticode-Signatur. Der Start von Studio wird verzögert, wenn Windows die Signatur nicht überprüfen kann.) Als Workaround können Sie das Überprüfen der Authenticode-Signatur deaktivieren, wie unter <http://support.citrix.com/article/CTX120115> beschrieben. [#337698]
- Wenn Sie mit Maschinenerstellungsdiensten (MCS) Maschinen auf VMware vSphere bereitstellen, spiegelt die Kombination der CPU-Sockets und Kernspeicher auf den bereitgestellten Maschinen die Kombination der CPU-Sockets und Kernspeicher auf dem Basisimage wider, das für die Maschinenkatalogerstellung verwendet wird. Wenn während der MCS-Katalogerstellung die Anzahl der ausgewählten virtuellen CPUs höher ist als die maximal mögliche auf dem Host, werden die bereitgestellten Maschinen mit der maximal möglichen Anzahl Sockets und Kernspeicher für diesen Host erstellt, ohne dass der Benutzer darauf hingewiesen wird. [#331269]
- Wenn Sie System Center Virtual Machine Manager in einer reinen IPv6-Umgebung verwenden und Maschinenkataloge mit den Maschinenerstellungsdiensten (MCS) erstellen, haben alle VMs IPv4 und IPv6, selbst wenn die Master-VM ohne IPv4 im TCP/IP-Stapel konfiguriert ist. Dies ist ein Problem mit Microsoft und es gibt kein Workaround. [#371712]
- Wenn Sie mit VMware vSphere in einer IPv4- und IPv6-Umgebung mit VMware ESX Hypervisoren verwenden, die mit VMXNET3-Netzwerkadaptern konfiguriert sind, haben alle VMs IPv4 und IPv6, selbst wenn die Master-VM im TCP/IP-Stapel ohne IPv4 konfiguriert ist. Dies ist ein Problem mit VMware und es gibt kein Workaround. [#371712]
- Lange Maschinenkatalognamen und lange Speicherpfadnamen führen möglicherweise zu einem Datenträgeranbindungsfehler im VMM-Auftragsfenster. Microsoft hat eine Beschränkung von maximal 255 Zeichen für die Länge der Dateipfad für VM-Ressourcen festgelegt. Dieses Problem tritt bei lokalen Speichern auf einem eigenständigen Hyper-V-Host auf, wegen der langen Dateipfade zum Speichern von VMs, ist jedoch nicht auf eigenständige Hyper-V-Hosts beschränkt. [#359673]

Lösungsansatz:

- Erstellen Sie VMM-MCS-Kataloge mit kurzen Namen, insbesondere dann, wenn über einen langen Pfad auf den Speicher zugegriffen wird.
- Kürzen Sie den Pfad zum Speicher, in dem die VMs gespeichert sind.
- Um zu einer zuvor genutzten Datenbank zu wechseln, müssen Sie das SDK verwenden; dies kann nicht mit Studio erfolgen. [#355993]

Wechseln zur Datenbank für die Konfigurationsprotokollierung

- Set-LogDbConnection -DataStore 'Logging' -DBConnection \$null
 - Set-LogDbConnection -DataStore 'Logging' -DBConnection '<neue Datenbankverbindungszeichenfolge>'
- Wechseln zur sekundären Überwachungsdatenbank
- Set-MonitorDbConnection -DataStore 'Monitor' -DBConnection \$null
 - Set-MonitorDbConnection -DataStore 'Monitor' -DBConnection '<neue Datenbankverbindungszeichenfolge>'

Beispielsweise kann die neue Datenbankverbindungszeichenfolge sein:

'Server = dbserver; Initial Catalog = dbname; Integrated Security = True'

- Wenn Sie Maschinen auf der Hypervisorplattform mit Maschinenerstellungsdiensten bereitstellen, werden CPUs aber keine Kernspeicher hinzugefügt. Wenn Sie den CPU-Wert bei der Katalogerstellung ändern, werden für die Produkte, die pro CPU lizenziert sind, möglicherweise weitere Lizenzen benötigt. Beispiel: Das Masterimage hat eine CPU und vier Kernspeicher und Sie ändern den CPU-Wert während der Katalogerstellung.
- Wenn Sie bei der MCS-Katalogerstellung mehr CPUs wählen als maximal auf dem Host zugelassen, ist die Höchstanzahl der für diesen Host unterstützten bereitgestellten Maschinen erreicht und Sie werden nicht über die Differenz benachrichtigt. Beispielsweise kann ein Windows-Desktopbetriebssystem nur zwei physische CPUs nutzen; somit werden nur zwei angezeigt, selbst wenn mehr zugewiesen sind.

Stellen Sie als Workaround sicher, dass die Masterimage-VM die gleiche virtuelle CPU-Konfiguration hat, die Sie für den Katalog bereitstellen möchten. [#331274]

- Wenn der Lync 2013-Client von einer Desktopbetriebssystemmaschine oder einer Serverbetriebssystemmaschine bereitgestellt wird, funktioniert das Videochat-Feature nicht. Informationen zur Verwendung von Lync 2013 finden Sie im Artikel zur Unterstützung für das Microsoft Lync 2013 VDI Plug-In durch XenDesktop 7.x, XenApp 6.x und Citrix Receiver 4.x (<http://support.citrix.com/article/CTX138408>). [#371818]
- Beim Starten einer Seamlessanwendung auf einem Windows-Server hat das Fenster ggf. nicht das Aero-Design, selbst wenn Enhanced Desktop Experience aktiviert ist.
 - Benutzer, die nur Seamlessanwendungen starten, erhalten nie das Aero-Design.
 - Benutzer mit Seamlessanwendungen und Desktops erhalten das Aero-Design nach dem Herstellen der ersten Desktopsitzung; Seamlessanwendungen haben danach das Aero-Design.
 Stellen Sie als Workaround das Citrix Enhanced Desktop-Design im Standardbenutzerprofil ein, sodass alle Benutzer auf diesem VDA Enhanced Desktop Experience für die Seamlessanwendungen erhalten. Das Design ist Teil der VDA-Installation und muss für alle VDAs eingestellt werden. [#348812]
- Von Studio an eine Windows 8-Maschine gesendete Nachrichten werden auf dem Windows 8-Desktop statt des Windows-Standarddisplays (früher Metro genannt) angezeigt. Der Benutzer muss zum Desktopmodus wechseln, um die Nachricht anzuzeigen. Dies ist ein Drittanbieterproblem und es gibt kein Workaround. [#387356]
- Wenn eine Site mehr als ein Hostinginfrastruktur-Objekt mit dem gleichen Namen enthält, wird Studio möglicherweise nicht richtig angezeigt. Beim Erstellen von Hostinginfrastruktur Objekte (z. B. Netzwerke und Speicher), sollten Sie einen eindeutigen Namen angeben. [#384959]
- Wenn Sie einen Administrator mit einem neuen Geltungsbereich mit der delegierten Administration erstellt haben, müssen Sie das Studio-Display aktualisieren. Sonst erhalten Sie u. U. einen Berechtigungsfehler, wenn Sie sich als der neue Administrator anmelden und versuchen, eine neue Verbindung oder Ressource zu erstellen. [#386634]
- Wenn Sie mit dem PowerShell SDK einen Sitenamen ändern (set-configsite -sitename "Namen ändern"), erhalten Sie eine Fehlermeldung, wenn Sie versuchen, der Site einen anderen Controller hinzuzufügen. Als Workaround sollten Sie die Site auf den Originalnamen umbenennen (entsprechend ServiceGroupName auf der Ausgabe von Get-ConfigRegisteredServiceInstance), bevor Sie den neuen Controller der Site hinzufügen. Danach können Sie die Site wieder umbenennen. [#386919]
- Auf Desktopbetriebssystemmaschinen gehostete Anwendungen mit zufälliger Zuweisung werden möglicherweise nicht geöffnet, nachdem das Dialogfeld "Laden" ausgeblendet wird. Das Problem tritt auf, wenn der Start der Anwendung länger dauert als das standardmäßige Timeout von einer Minute und die Sitzung automatisch beendet wird. [#389025]
Ändern Sie als Administrator das Basisimage und stellen es mit einem geänderten Timeoutwert wie folgt neu bereit:

Suchen Sie die Registrierungsschlüssel:HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshell\TWI (erstellen Sie den Schlüssel, wenn er nicht verfügbar ist.)

Name: ApplicationLaunchWaitTimeoutMS

Typ: REG_DWORD

Wert: Zusätzlich benötigter Timeout in Millisekunden

Hinweis: Wenn Sie einen Wert angeben, der kleiner als 10.000 ist, wird 10.000 verwendet, da 10 Sekunden der Mindestwert für die Überschreibung ist.

- Wenn sich XenServer-Hosts in einem Pool befinden und eine sichere Verbindung von diesem Produkt mit dem Pool hergestellt wird, erhalten Sie ggf. die folgende Fehlermeldung: "Ungültige Verbindungsadresse". Stellen Sie sicher, dass die Adresse gültig ist, und dass sie auf einem Host im XenServer-Pool verweist. Sie vermeiden dieses Szenario, wenn Sie Zertifikate, die von einer vertrauenswürdigen Stammzertifizierungsstelle ausgestellt sind, auf allen XenServer-Hosts im Pool installieren. [#392279]
- Wenn Sie einen Maschinenkatalog mit den Maschinenerstellungsdiensten erstellen, kann ein Datenbankfailover auftreten, wenn Sie das Masterimage kopieren oder eine Maschine hinzufügen. Die folgende Meldung angezeigt: Die Sammlung wurde geändert, Auflistung kann u. U. nicht ausgeführt werden. Stellen Sie sicher, dass die Datenbank funktionsfähig ist, bevor Sie die folgenden Aktionen ausführen. Bei einem Failover:
Löschen Sie beim Kopieren des Masterimages den Maschinenkatalog und erstellen ihn neu. Wählen Sie beim Hinzufügen von Maschinen den MCS-Maschinenkatalog aus und klicken Sie auf "Maschinen hinzufügen". Sollte der Fehler weiterhin auftreten, löschen Sie den Maschinenkatalog und erstellen Sie ihn neu. [#393889]
- Zum Bearbeiten von Hostverbindungen in Studio müssen Sie die Rolle "Volladministrator" oder die Rolle "Maschinenkatalog-Administrator" und "Alle Geltungsbereiche" verwenden. Ohne "Alle Geltungsbereiche" tritt u. U. ein unerwarteter Fehler auf. [#392799]
- Beim Aktualisieren eines Maschinenkatalogs werden trotz eines erfolgreichen Updates keine Benachrichtigungsmeldungen auf Sitzungen angezeigt, die auf Maschinen in diesem Katalog ausgeführt werden. Als Lösungsansatz kann der Administrator versuchen, die Nachricht erneut über den Knoten "Suchen" in Studio zu senden. [#390726]
- Zum Durchführen der Discovery und zum Einrichten von Anwendungen in Bereitstellungsgruppen sollten Sie sicherstellen, dass einige Maschinen registriert sind und eingeschaltet werden können. Für die Discovery einer App-V-Anwendung müssen Sie die App-V-Veröffentlichung in Studio konfigurieren. [#393676]
- Wenn ein Benutzer die erste App-V-Anwendung startet, schlägt ein zu schnelles Starten einer zweiten App-V-Anwendung u. U. fehl. Wenn dieses Problem auftritt, sollte der Benutzer ein paar Minuten warten, damit die anfängliche Synchronisierung abgeschlossen werden kann, und dann die zweite Anwendung erneut starten. Weitere Informationen zu diesem Problem finden Sie unter [CTX138056](#). [#397521]
- Wenn eine physische Maschine neu für Remote-PC-Zugriff konfiguriert wurde, kann die erste Registrierung an einer XenDesktop-Site fehlschlagen. Nach etwa fünf Minuten kann die Maschine neu registriert werden. Vor der erfolgreichen Neuregistrierung können Sie keine Verbindung über StoreFront mit der Maschine herstellen. [#394791]
- Wenn Sie versuchen, eine Verbindung zu Microsoft System Center Virtual Machine Manager (VMM) auf einem Delivery Controller herzustellen, auf dem die VMM-Konsole installiert ist, warnt Studio nicht, dass die Verbindung nur richtig funktioniert, wenn die VMM-Konsole auf allen Controllern in der Site installiert ist. Installieren Sie die VMM-Konsole auf alle Controllern und starten Sie diese neu, um dieses Problem zu vermeiden. [#348614]
- Wenn das Antivirenprogramm BitDefender auf einem VDA installiert ist, können Sie möglicherweise keine Maschinenkataloge erstellen. Für dieses Problem gibt es kein Workaround. [#392705]
- Wenn XenServer 6.2 oder 6.1 mit einem offenen vSwitch-Controller konfiguriert wurde, erstellen die Maschinenerstellungsdienste nur dann Maschinen, wenn die Option Das Masterimage wird mit Windows XP oder Windows Vista erstellt aktiviert ist. Bei Auswahl dieser Option stehen einige Produktfeatures nicht für diesen Katalog zur Verfügung. [#394771]
- Wenn ein Citrix CloudPlatform-Administrator versucht, einen Maschinenkatalog auf einem System zu erstellen, auf dem mindestens ein Benutzerkonto auch XenDesktop verwendet, schlägt die Erstellung möglicherweise fehl. Zur Problemumgehung sollte der Administrator ein Benutzerkonto erstellen und dieses verwenden, um Maschinen zu erstellen. [#452528]
- Wenn Sie Maschinenkataloge erstellen, für die das Provisioning mit den Maschinenerstellungsdiensten (MCS) durchgeführt wurde und die sehr viele Maschinen enthalten, wird beim Löschen von Maschinen oder bei Energieaktionen möglicherweise der verfügbare Speicher von Studio aufgebraucht und sie schlagen fehl auf Delivery Controllern mit 8 GB RAM oder weniger. Nachdem Sie Aufgaben mit einer großen Anzahl von Maschinen ausgeführt haben, löschen Sie auf der Registerkarte "Aktionen" die Hintergrundaktionen, bevor Sie Studio schließen und neu starten. Informationen über eine Lösung für dieses Problem finden Sie unter [CTX140706](#). [#455848]
- Nach dem Upgrade auf XenDesktop 7.5 können Sie keine Verbindung mit CloudPlatform oder Amazon Web Services (AWS) erstellen, wenn für die Site eine Verbindung mit einem lokalen Hypervisor konfiguriert ist. Sie müssen eine neue 7.5-Site erstellen. Bei einer Hybridbereitstellung, die Cloudsites und nicht cloudbasierte Sites umfasst, müssen Sie separate Sites erstellen, die dieselbe StoreFront-Site verwenden. [#454114]

Probleme mit Director

- Wenn Sie virtuelle Windows XP-Desktops überwachen, auf denen WinRM 2.0 für Benutzer mit VDAs vor 7 ausgeführt werden, müssen Sie die WinRM-Port-Abhörreihenfolge ändern. Weitere Informationen zum Ändern der Einstellung auf 5985,80 finden Sie unter [Erweiterte Konfiguration](#). [#273609]
- Die Konfiguration von Benutzern mit Anmeldeskripten über Active Directory-Benutzer und -Computer schlägt fehl und Daten werden nicht im Bereich "Anmeldedauer" angezeigt. Verwenden Sie zum Konfigurieren von Benutzern mit Anmeldeskripten eine Gruppenrichtlinie. [#393259]
- Die Daten zur Anmeldedauer von einer Erstanmeldung an Personal vDisk-VDA werden in Director u. U. nicht gesammelt oder angezeigt. Für weitere Anmeldungen werden die Daten normal angezeigt. [#383941]
- Wenn Sie eine Konsolensitzung von einem Windows Server 2012-Desktop aus ausführen, kann das Navigieren zur Benutzerdetailseite in Director zu einem Fehler führen, sodass keine Daten

angezeigt werden.

Um dieses Problem zu umgehen, registrieren Sie den VDA, melden Sie sich bei der aktuellen Sitzung ab und melden Sie sich dann erneut beim VDA an. [#388513]

- Im Bereich "Infrastruktur" der Seite "Dashboard" von Director wird für Citrix CloudPlatform-basierte Hostverbindungen, Amazon Web Services, Hyper-V und Microsoft System Center Configuration Manager "Nicht verfügbar" ohne Statusinformationen angezeigt. [#449806, #446397]
- Die Anmeldedauer wird im Dashboard von Director nicht aktualisiert, wenn Benutzer eine gehostete Sitzung starten. [#386860]

Probleme mit HDX

Hinweis: Aktuelle Updates für HDX Flash-Kompatibilität finden Sie unter [CTX136588](#).

- Bei GPU-Passthrough und NVIDIA Kepler-Karten schlägt der erste Verbindungsversuch möglicherweise fehl, wenn Sie ein HDX 3D Pro-Benutzergerät mit drei oder vier Bildschirmen verwenden. Versuchen Sie in diesem Fall erneut, eine Verbindung herzustellen. [#422049]
 - Obwohl Webcams möglicherweise die H.264-Komprimierung unterstützen, wird die Hardwarekomprimierung in diesem Release nicht unterstützt. Sie müssen daher die Softwarekomprimierung für diese Webcams verwenden. Fügen Sie hierzu auf Benutzergeräten einen Registrierungseintrag unter HKEY_CURRENT_USER\Software\Citrix\HdxRealTime hinzu; fügen Sie den DWORD-Registrierungsnamen DeepCompress_ForceSWEncode hinzu. Bei der Einstellung "1" wird die Softwarekomprimierung verwendet. Standardmäßig ist diese Einstellung deaktiviert und die Hardwarekomprimierung wird verwendet. [#357356]
 - Das HDX RealTime-Webcamvideo ist verzögert, wenn die Videoauflösung höher als 720 p (1280 x 720) ist. [#350187]
 - Bei der ständigen Verwendung der HDX Flash-Umleitung reagiert die Sitzung u. U. nicht mehr. [#350085, 361926]
 - HDX-Webcam unterstützt die meisten unformatierten Webcamformate, die eine Webcam unterstützt; in seltenen Fällen, wenn die Webcam ein nicht unterstütztes Format hat, funktioniert die Webcam möglicherweise nicht wie die Citrix HDX-Webcam. [#338318]
 - Beim Verwenden von Citrix HDX-Webcam mit gewissen Webcammodellen werden manchmal mehrere Duplikatbilder angezeigt. [#367322]
 - HDX RealTime-Webcams werden für die folgenden Anwendungen nicht unterstützt:
 - Citrix GoToMeeting, wenn es auf Serverbetriebssystemmaschinen mit Windows 2012-Betriebssystemem gehostet wird. [#346430]
 - GoToMeeting (auf jeder Plattform), wenn die Webcam nach Beginn der Besprechung angeschlossen wird. [#346140]
 - Microsoft Lync 2013 und Adobe Connect mit VDAs unter Windows 8-, Windows 8.1- und Windows 2012-Betriebssystemen. [#340784, 348506, 459732]
 - Microsoft Office Communications Server-Videoanrufe (OCS), wenn die Webcam angeschlossen wird, wenn der Anruf bereits ausgeführt wird. [#370236]
 - Microsoft Silverlight. Dies ist ein zeitweiliges Problem. Um das Problem zu umgehen, aktivieren Sie auf dem Benutzergerät den Legacy-Codec, indem Sie einen DWORD-Registrierungsschlüsselnamen EnableDeepcompress_Client unter HKEY_CURRENT_USER\Software\Citrix\HdxRealTime hinzufügen und auf 0 stellen. [#379779]
 - 64-Bit-Videoanwendung. Videokomprimierung für 64-Bit-Anwendungen wird nicht unterstützt. [#366515]
 - Bei der Verwendung von HDX 3D Pro mit dem XenDesktop 5.6 Feature Pack 1 VDA auf virtuellen Windows XP-Desktops ist das Kontrollkästchen "Fine Drawing (2D)" bei der ersten Verbindung ausgewählt und manchmal abgeblendet. Ursache sind Verzögerungen bei Registrierungsupdates, sodass während der Initialisierung in der Benutzeroberfläche des Konfigurationstools falsche Standardwerte angezeigt werden. [#353031]
- Als Workaround können Sie die Sitzung trennen und neu verbinden.

Sollte das Problem weiterhin auftreten, löschen Sie vorherige Sitzungsinformationen und löschen Sie die Einstellungen im folgenden Registrierungsschlüssel:

HKEY_CURRENT_USERS\Software\Citrix\HDX3D\BitmapRemotingConfig

- Unter Windows XP wird Citrix HDX-Webcam u. U. nicht erkannt. Installieren Sie Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package über <http://www.microsoft.com/de-de/download/details.aspx?id=14431> und versuchen Sie es erneut, um dieses Problem zu beheben. [#382733]
 - Beim Anzeigen des Knotens "Grafikkarten" in der Geräte-Manager-Konsole wird die Standard-VGA-Grafikkarte in der Liste mit einem gelben Hinweissymbol angezeigt. Sie können diese Warnung ignorieren, da sie sich nicht auf die Funktionalität auswirkt. Diese Warnung wird angezeigt, da ein Legacymodell des XPDM-Anzeigetreibers (Standard-VGA-Grafikkarte) nicht geladen werden kann, wenn ein neuer Anzeigetreiber vom Typ WDDM (Citrix Anzeigetreiber) installiert ist. [#339390]
 - Benutzer stellen u. U. Probleme bei der Wiedergabe von Mediendateien auf Windows 8-Benutzergeräten fest. Dies liegt daran, dass das Standardprogramm für clientseitige Inhaltsabrufprotokolle, mit denen Mediendateien auf Benutzergeräte gestreamt werden, von diesem Produkt nicht registriert wird. Um dieses Problem für die Protokolle Microsoft Media Streaming (MMS) und Real Time Streaming (RTS) zu umgehen, ändern Sie das Standardprogramm für die Wiedergabe von Mediendateien von Windows Media Player zu Citrix CSF Handler. Für HTTP (HyperText Transfer Protocol) ist kein Workaround möglich. [#328805]
 - Die TWAIN-Umleitung schlägt für gehostete, freigegebene Desktops und Anwendungen fehl. Dies ist ein Drittanbieterproblem bezüglich TWAIN-Anwendungen, die benötigen, dass bestimmte TWAIN-Binärdateien in bestimmten Pfaden sein müssen. [#300854, 340999]
- Kopieren Sie als Workaround auf der Windows Server 2012-Maschine mit dem VDA diese Dateien an die folgenden Speicherorte:
- Kopieren Sie die Datei "twain_32.dll" in das Verzeichnis "\WINDOWS" des Benutzerprofils (z. B. Datei twain_32.dll in den Ordner "%USERPROFILE%\Windows").
 - Kopieren Sie die Datei "twain_32.dll.mui" in das Verzeichnis "\WINDOWS\en-US\" des Benutzerprofils (z. B. Datei twain_32.dll.mui in den Ordner "%USERPROFILE%\Windows\en-US").
 - Die 64-Bit-Version von Windows Media Player oder QuickTime Player kann einige Videodateien nicht mit der serverseitigen Wiedergabe wiedergeben, wenn die HDX MediaStream Windows Media-Umleitung deaktiviert ist. Verwenden Sie als Workaround die 32-Bit-Version von Windows Media Player. [#384759]
- Im virtuellen Desktop ausgewählte universelle Druckserver-Drucker werden im Fenster "Geräte und Drucker" in der Windows-Systemsteuerung nicht angezeigt. In den Anwendungen stehen diese Drucker den Benutzern jedoch zum Drucken zur Verfügung. Dieses Problem tritt nur auf Windows Server 2012- und Windows 8- Plattformen auf. [#335153]
 - Bei GPU-Passthrough und NVIDIA Kepler-Karten schlägt der erste Verbindungsversuch möglicherweise fehl, wenn Sie ein HDX 3D Pro-Benutzergerät mit drei oder vier Bildschirmen verwenden. Der zweite Verbindungsversuch sollte erfolgreich sein. [#422049]
 - Der Windows-Computer eines Benutzers reagiert nicht mehr, wenn eine GoToMeeting-Sitzung in einer Remote-PC-Zugriff-Sitzung mit einem Computer mit Intel Core i7-Prozessor gestartet wird und eine für USB-Umleitung konfigurierte Webcam verwendet wird. Wenn dieses Problem auftritt, starten Sie den Computer des Benutzers und die Remote-PC-Zugriff-Sitzung neu. Die Sitzung wird fortgesetzt, wo die Verbindung unterbrochen wurde. Sie vermeiden dieses Problem, indem Sie HDX-Video komprimierung anstelle von USB-Umleitung verwenden. Weitere Informationen finden Sie unter <http://support.citrix.com/proddocs/topic/xendesktop-7/hd-new-graphics-video.html>. [# 423284]
 - Benutzergeräte, auf denen Receiver für HTML5 ausgeführt wird, können möglicherweise keine Verbindung zu einer Serverbetriebssystemmaschine herstellen, auf der Windows Server 2012 R2 ausgeführt wird. So vermeiden Sie dieses Problem:
 - Verwenden Sie eine vorhandene Maschine als die Windows-Serverbetriebssystemmaschine statt einer Maschine, die mit den Maschinenerstellungsdiensten oder Provisioning Services erstellt wurde.
 - Wenn Sie mit den Maschinenerstellungsdiensten erstellte Maschinen verwenden, ändern Sie die Registrierung auf dem Masterimage für den Katalog und erstellen Sie folgenden DWORD-Wert: Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab.
HKEY_LOCAL_MACHINE\Software\Citrix\GroupPolicy\Defaults\lcaPolicies\AcceptWebSocketsConnections = 1 HKEY_LOCAL_MACHINE\Software\Citrix\GroupPolicy\Defaults\lcaPolicies\AllowC Da die WebSockets-Portnummer durch Bearbeiten der Registrierung festgelegt wird, brauchen Sie die Citrix Richtlinie für WebSockets-Verbindungen für diesen Katalog nicht zu aktivieren.
 - Wenn Sie eine Maschine verwenden, die mit Provisioning Services erstellt wurde, folgen Sie den Empfehlungen unter [CTX139265](#). [#424064]

- Wenn HDX Flash-Umleitung in einer Konfiguration mit zwei Bildschirmen verwendet wird und das Fenster mit dem Flash-Inhalt im Vollbildmodus auf einem Bildschirm angezeigt wird, kann das Klicken auf eine beliebige Stelle auf dem Bildschirm dazu führen, dass das Fenster mit dem Flash-Inhalt den Fokus verliert und hinter das Sitzungsfenster verschoben wird. Dieses Verhalten ist bei HDX Flash-Umleitung beabsichtigt. In diesem Fall können Benutzer das Fenster mit dem Flash-Inhalt wieder sichtbar machen, indem sie den Modus für das Sitzungsfenster mit der Citrix Desktop Viewer-Symbolleiste vom Vollbild- in den Fenstermodus ändern. [#567132]

Lizenzierungsprobleme

- Wenn die Lizenzierung deinstalliert und neu installiert wird und ein Produktadministrator mit Lesezugriff versucht, die Lizenzinformationen in Studio anzuzeigen, wird der Fehler "Sie haben nicht die notwendige Berechtigung, um diesen Vorgang auszuführen" angezeigt. Ein Administrator, der nur Lesezugriff hat, hat keine Berechtigung, den Lizenzserver als vertrauenswürdig anzusehen. Als Workaround muss ein Volladministrator für die Lizenzierung den Lizenzserver im Knoten "Lizenzierung" authentifizieren. [#380982]
- Wenn Sie den Lizenzserver installiert haben, ohne ihn nach der Installation erfolgreich (mit dem Lizenzserverkonfigurationstool) zu konfigurieren, schlagen alle nachfolgenden Lizenzserverupgrades fehl. Stellen Sie als Workaround sicher, dass jede Lizenzserverinstallation anschließend mit dem Lizenzserver-Konfigurationstool konfiguriert wird. [#377079]
- Beim Start der License Administration Console oder des Simple License Service wird möglicherweise eine leere Seite angezeigt, wenn in Internet Explorer die verstärkte Sicherheitskonfiguration aktiviert ist und die License Administration Console bzw. Simple License Service nicht zu den vertrauenswürdigen Sites gehören. Workaround: Deaktivieren Sie die verstärkte Sicherheitskonfiguration für Internet Explorer [#382429]
- Wenn der Port 8083 während der Produktinstallation oder dem Produktupgrade in Verwendung ist, schlagen die Lizenzserverkonfiguration und die Installation mit einem Lizenzserverkonfigurationsfehler fehl. Überprüfen Sie als Workaround das Ereignisprotokoll, um sicherzustellen, dass es sich wirklich um den Fehler "Port wird verwendet" handelt. Ist dies der Fall:
 1. Doppelklicken Sie zum Deinstallieren des Lizenzservers auf die Datei CTX_Licensing.msi im Ordner "x64\Licensing" auf dem Produktinstallationsmedium.
 2. Führen Sie das Installationsprogramm erneut aus. Einige Komponenten werden als teilweise installiert angezeigt. Klicken Sie auf "Installieren" und das Installationsprogramm schließt die Installation ab und konfiguriert die erforderlichen Produktkomponenten.
 3. Installieren Sie den Lizenzserver manuell und geben Sie Portnummern an, die nicht von anderen Anwendungen auf der Maschine verwendet werden. [#390815]
- Die Benutzerliste für die License Administration Console und die Webseite für den Citrix Simple License Service unterstützt keine Nicht-ASCII-Zeichen in Benutzer-/Gruppennamen. Aufgrund dieser Einschränkung wird auf einem russischen Betriebssystem die Gruppe VORDEFINIERT\Administratoren nicht der Benutzerliste hinzugefügt, weil sie Nicht-ASCII-Zeichen enthält. Dies gilt für Neuinstallationen und Upgrades. Benutzer in der Gruppe VORDEFINIERT\Administratoren in einem früheren Release von XenDesktop und dem Simple License Service haben nach dem Upgrade keinen Zugriff auf die License Administration Console oder den Simple License Service.
 - Als Workaround fügen Sie nach der Installation die russischen Benutzer-/Gruppennamen in einer ASCII-Zeichen-Version über die License Administration Console hinzu. Alternativ installieren Sie den Lizenzserver auf einem der anderen unterstützten Betriebssysteme. [#395305]
- Wenn Sie das Produkt installieren oder aktualisieren und permanente Lizenzen haben, werden die Lizenzen in Studio u. U. mit dem Ablaufdatum 01.01.2000 angezeigt. Sie können dieses Ablaufdatum ignorieren und die Desktops und Anwendungen starten. [#402975]

Probleme beim lokalen App-Zugriff

- Wenn der lokale App-Zugriff in einer Windows 8- oder Windows Server 2012- Hostumgebung aktiviert ist, und die Erweiterung für eine auf dem Client gehostete App keine Dateitypzuordnung hat, schlägt die Umleitung der Dateitypzuordnung fehl. Der Benutzer wird aufgefordert, "Suchen Sie eine App im Store" oder "Weitere Optionen" auszuwählen. [#372834]
 - Verwenden Sie als Workaround eine dieser Methoden auf dem VDA-Masterimage:
 - Benennen Sie den Registrierungswert DelegateExecute für HKEY_CLASSES_ROOT\Unbekannter\Shell openas\\DelegateExecute um.
 - Öffnen Sie eine Datei mit der Erweiterung im Editor. Die Umleitung der Dateitypzuordnung funktioniert bei weiteren Versuchen.
- Die URL-Umleitung ist standardmäßig von Microsoft auf Windows Server 2012 deaktiviert. Um sie zu aktivieren, deaktivieren Sie die verstärkte Sicherheitskonfiguration für Internet Explorer. [#356260]
- Änderungen an den lokalen App-Zugriff-Eigenschaften in einer Sitzung werden nicht automatisch übernommen. Als Workaround melden Sie sich ab und wieder an. [#357488]
- Wenn lokale App-Zugriff-Anwendungen auf Windows 8- oder Windows 2012-Plattformen gestartet werden können diese VDAs nicht von der Modern Shell gestartet werden. Workaround: Schließen Sie die lokale Anwendung und starten Sie dann die VDA-Anwendung. [#359670]
- Shellhook.dll wird nicht mit Receiver für Windows 3.4 und früheren Versionen geladen, wenn lokale Anwendungen gestartet werden. Ändern Sie als Workaround den Wert des Registrierungsschlüssels LocalApplnit_DLLs unter HKLM\Software unter Microsoft Windows NT\\CurrentVersion\Windows auf 1. Wenn der Workaround nicht verwendet wird, funktioniert die Umleitung der Dateitypzuordnung vom Client zum Server nicht richtig. [#356130]
- Die URL-Umleitung schlägt u. U. Websites fehl, für die ein Popublocker aktiviert ist. Um dieses Problem zu umgehen, deaktivieren Sie den Popublocker. Aus Sicherheitsgründen wird das Deaktivieren von Popublockern nicht empfohlen. [#371220]
- Wenn der Aero-Modus für eine VDA-Sitzung aktiviert ist, bestehen Inkonsistenzen zwischen den lokalen VDA-Anwendungen und den auf dem Client gehosteten Anwendungen (z. B. ALT + Tab; blinkende Einträge auf der Taskleiste, Sprungliste, Echtzeitschau). Deaktivieren Sie den Aero-Modus für Kompatibilität. [#361043]
- Die Flash-Umleitung hat Kompatibilitätsprobleme, z. B. einen schwarzen WMP-Bildschirm und ein Flash-Pseudocontainer außerhalb des Bereichs. Als Workaround sollten Sie die Flash-Umleitung deaktivieren. [#360182]
- Die DesktopgestaltungsUmleitung für Grafiken ist deaktiviert, wenn der lokale App-Zugriff aktiviert ist. Deaktivieren Sie den lokalen App-Zugriff, um dieses Feature zu verwenden. [#377386]
- Wenn die Sitzungszuverlässigkeit und der lokale App-Zugriff aktiviert sind und die Netzwerkkonnektivität unterbrochen wird, wenn Sie auf der Produktsymbolleiste auf "Home" klicken und den Desktop des Clients anzeigen, wird in der VDA-Sitzung u. U. nicht der letzte Bildschirm vor der Unterbrechung der Netzwerkkonnektivität angezeigt. Stattdessen werden nur die Produktsymbolleiste und der Desktop des Clients angezeigt. Auf der Symbolleiste funktioniert nur die Option "Trennen". [#357769]
- Wenn der lokale App-Zugriff aktiviert ist und ein Benutzer vor oder während der Wiedergabe einer Mediendatei für die Sitzung Vollbild aktiviert, wird das Bild nur teilweise oder gar nicht angezeigt. Starten Sie die Sitzung neu, um das Problem zu beheben. [#402702]
- Das Starten von clientgehosteten Verknüpfungen im lokalen Programmordner in einer Sitzung, deren Verbindung von japanischen oder chinesischen Clients erstellt wurde, kann zu einem Anwendungsfehler führen. Melden Sie sich von der Sitzung ab und starten Sie die Sitzung neu, um dieses Problem zu lösen. Wenn das Problem weiterhin besteht, erstellen Sie eine clientgehostete Anwendung wie unter [Bereitstellen von Zugriff auf lokale Anwendungen](#) erläutert. Auf diese Weise veröffentlichen Sie bestimmte lokale Anwendungen anstelle von clientgehosteten Verknüpfungen. [# 399100]

Probleme bei Personal vDisk

- Das Vorhandensein von Antivirenprodukten kann sich darauf auswirken, wie lange das Durchführen einer Bestandsaktualisierung oder eines Updates dauert. Die Leistung kann verbessert werden, wenn Sie CtxPvD.exe und CtxPvD.Svc.exe der PROCESS-Ausschlussliste des Antivirenprodukts hinzufügen. Diese Dateien befinden sich im Ordner C:\Programme\Citrix\personal vdisk\bin. [#326735]
- Wenn copy-on-write aktiviert ist und alternative Datenströme verwendet werden, werden Änderungen am Inhalt der alternativen Datenströme nicht gespeichert. Copy-on-write unterstützt keine alternativen Datenströme. Sie vermeiden dieses Problem, wenn Sie die Dateien mit den alternativen Datenströmen auf das Gastvolume verlagern. Weitere Informationen finden Sie in den benutzerdefinierten Regeldateien in %ProgramData%\Citrix\personal vDisk\Config mit benutzerdefinierten_* Namen. [#369193]
- Feste Links zwischen Dateien, die vom Masterimage geerbt wurden, bleiben in persönlichen vDisk-Katalogen nicht erhalten. [#0368678]
- Nach dem Upgrade von Office 2010 auf 2013 auf dem Personal vDisk-Masterimage kann Office nicht auf virtuellen Maschinen gestartet werden, da der KMS-Lizenzierungsproduktschlüssel von Office beim Upgrade entfernt wurde. Als Workaround können Sie Office 2010 deinstallieren und Office 2013 auf dem Masterimage neu installieren. [#391225]
- Personal vDisk-Kataloge unterstützen keine VMware Paravirtual SCSI (PVSCSI)-Controller. Verwenden Sie den Standardcontroller, um dieses Problem zu vermeiden. [#394039]
- Für virtuelle Desktops, die mit Personal vDisk Version 5.6.0 erstellt wurden und auf 7 aktualisiert werden, finden Benutzer, die sich an der Master-VM anmelden, nicht alle Dateien in der gepoolten VM. Dieses Problem tritt auf, da ein neues Benutzerprofil erstellt wird, wenn sich Benutzer an der gepoolten VM anmelden. Für dieses Problem gibt es keinen Workaround. [#392459]

- Personal vDisks, die unter Windows 7 ausgeführt werden, können das Feature "Sichern und Wiederherstellen" nicht verwenden, wenn der Windows-Systemschutz aktiviert ist. Wenn der Systemschutz deaktiviert ist, wird ein Backup des Benutzerprofils aber nicht der Datei userdata.v2.vhd erstellt. Citrix empfiehlt, den Systemschutz zu deaktivieren und ein Backup des Benutzerprofils mit dem Feature "Sichern und Wiederherstellen" zu erstellen. [#360582]
- Beim Erstellen einer VHD-Datei auf der Basis-VM mit dem Datenträgerverwaltungstool können Sie die virtuelle Festplatte (VHD) möglicherweise nicht bereitstellen. Um dieses Problem zu umgehen, kopieren Sie die VHD auf das PvD-Volumen. [#355576]
- Nach der Aktualisierung eines Images auf einem virtuellen Desktop kann manchmal die persönliche vDisk, die dem Desktop zugeordnet ist, nicht gestartet werden, und eine Fehlermeldung (Fehlercode 0x10) wird angezeigt. Für dieses Problem gibt es keinen Workaround. [#416128]
- Office 2010-Verknüpfungen bleiben auf virtuellen Desktops erhalten, nachdem diese Software entfernt wurde. Um dieses Problem zu umgehen, löschen Sie die Verknüpfungen. [#402889]
- Wenn Sie Microsoft Hyper-V verwenden, können Sie mit Maschinen mit persönlichen vDisks keinen Katalog erstellen, wenn die Maschinen lokal gespeichert sind und die vDisks auf freigegebenen Clustervolumen (CSVs) gespeichert sind. Die Katalogerstellung schlägt fehl und ein Fehler wird angezeigt. Um dieses Problem zu umgehen, verwenden Sie einen anderen Speicherort für die vDisks. [#423969]
- Wenn Sie sich das erste Mal bei einem virtuellen Desktop anmelden, der von einem Provisioning Services-Katalog erstellt wurde, werden Sie aufgefordert, den Desktop neu zu starten, wenn die persönliche vDisk zurückgesetzt wurde (mit dem Befehl "ctxpvd.exe -s reset"). Um dieses Problem zu lösen, starten Sie den Desktop neu, wenn Sie dazu aufgefordert werden. Es handelt sich hier um eine einmalige Zurücksetzung, die bei erneuter Anmeldung nicht erforderlich ist. [#340186]
- Wenn Sie .NET 4.5 auf einer persönlichen vDisk installieren und .NET 4.0 bei einem späteren Update des Images installiert oder bearbeitet wird, schlagen Anwendungen, die von .NET 4.5 abhängig sind, fehl. Um dieses Problem zu umgehen, stellen Sie .NET 4.5 vom Basisimage aus als Imageupdate bereit.

Probleme mit Desktop Lock

- Nach der Installation von Citrix Desktop Lock auf einer in eine Domäne eingebundenen Windows 8-Maschine, wird nach dem Neustart u. U. ein schwarzer Desktophintergrund und statt des normalen Startbildschirms das Fenster "Arbeitsplatz" angezeigt. Geben Sie zum Abmelden von der Maschine Logoff in der Adressleiste des Fensters ein und drücken Sie die Eingabetaste. Weitere Informationen zum Konfigurieren von Windows 8-Maschinen mit Desktop Lock finden Sie unter [Konfigurieren von Desktop Lock](#). [#329075]
- Wenn eine Windows 8- oder Windows 7-Maschine ausfällt und der Benutzer versucht, die Maschine mit Strg+Alt+Entf neu zu starten, wird das Dialogfeld "Windows-Sicherheit" auf dem lokalen Desktop angezeigt. Dieses Problem tritt auch auf, wenn die Citrix Desktop Lock-Software installiert und konfiguriert ist. Um dieses Problem zu lösen, wählen Sie "Task-Manager starten", um das Dialogfeld "Neu starten" anzuzeigen. [#337507]
- Citrix Desktop Lock leitet Adobe Flash-Inhalte nicht an Benutzergeräte in der Domäne weiter. Der Inhalt kann angezeigt werden, wird aber auf dem Server wiedergegeben, nicht lokal. Als Workaround können Sie die Adobe Flash-Umleitung für serverseitigen Inhaltsabruf konfigurieren, um den Inhalt vom Server an das Clientgerät zu übergeben. Das Problem tritt nicht bei Geräten auf, die keiner Domäne angehören, und auch nicht, wenn der Inhalt mit Desktop Viewer angezeigt wird. [#263092]
- Der VDA-Start auf einem Benutzergerät kann fehlschlagen, wenn auf dem Gerät Windows 7 mit Desktop Lock und Receiver für Windows Enterprise 3.4 ausgeführt wird. Das Problem kann auftreten, wenn der Benutzer mit dem Internet verbunden ist oder keine Verbindung hat. Die Fehlermeldung "Keine Internetkonnektivität" wird dann ggf. angezeigt. Dies ist ein Drittanbieterproblem von Microsoft. Eine mögliche Lösung finden Sie unter <http://technet.microsoft.com/en-us/library/cc766017>. [#408642]

Endbenutzer- und VDA-Probleme

- Auf dem Benutzergerät können Handleprobleme mit wfica32.exe auftreten, wenn Windows Media Player-Dateien fortlaufend in einer Windows 32-Bit-XP-Clientsitzung wiedergegeben werden. [#378146]
- Die Schaltfläche "Trennen" steht nicht für die Benutzer zur Verfügung. Als Workaround können Sie eine Verknüpfung zum Hilfsprogramm TSDicon.exe bereitstellen, das im Betriebssystem enthalten ist. Benutzer können dann Sitzungen trennen. [#362937]
- Das Starten von VDA von einem Benutzergerät mit einem Smartcardleser kann fehlschlagen, wenn der Benutzer den VDA vorher von diesem Gerät gestartet und in Desktop Viewer "Trennen" ausgewählt hat. In diesem Fall wird dem Benutzer der Bildschirm "Smartcardanmeldeinformationen" oder die Meldung "Smartcard wird gelesen" angezeigt. Wählen Sie als Workaround eine der folgenden Optionen:
 - Nehmen Sie die Smartcard aus dem Leser heraus und stecken sie wieder ein.
 - Klicken Sie auf "Abbrechen". Drücken Sie dann im VDA Strg+Alt+Entf. [#322301]
- Benutzer haben möglicherweise nach der Trennung weiterhin Zugriff auf den Remotecomputer. Dies tritt auf, wenn der Benutzer den Befehl "Trennen" im Startmenü auswählt. Nach der Auswahl von "Trennen" entsteht eine Verzögerung. Wenn der Benutzer während dieser Verzögerung Strg+Alt+Entf drückt, steht der VDA auf dem Remotecomputer für mehrere Minuten weiterhin zur Verfügung, bevor die Verbindung getrennt wird. Während dieser Zeit hängt VDA auf dem Benutzergerät, bis VDA getrennt wird. [#322301]
- Bei VDAs, die auf Windows 7- und Windows 8-Plattformen installiert sind, werden u. U. zwei Mauszeiger angezeigt, ein beweglicher und ein auf der Benutzeroberfläche arretierter. Dies ist ein Drittanbieterproblem mit dem Nvidia-Treiber. Als Workaround können Sie die Nvidia GRID-Technologie (früher VGX) deaktivieren. Führen Sie MontereyEnable.exe -disable -reset aus und starten die Maschine neu. [#307921]
- Das Feature "Microsoft Desktopgestaltung" verursacht Skalierbarkeitskosten für VDAs. Für Benutzer, die maximale Skalierbarkeit benötigen, sollte die Desktopgestaltung für alle Benutzer, die die Desktopgestaltungsumgebung nicht verwenden, über eine Microsoft Richtlinie deaktiviert werden. [#386602]
- In bestimmten Szenarios wird die Sitzung wiederholt gesperrt, wenn Benutzer eine gesperrte Sitzung lokal entsperren. Dieses Problem kann auftreten, wenn der Benutzer eine Tastatur lokal abschließt oder ausschaltet, die in einer ICA-Remotesitzung mit einer Remote-PC-Zugriff-Maschine verbunden ist. Als Workaround sollten die Benutzer die Sitzung remote neu starten, die Sitzung trennen und sich dann erneut von der Konsole anmelden. [#382554]
- Wenn die Installation eines VDAs für Windows-Serverbetriebssysteme von der grafischen Benutzeroberfläche anscheinend hängen bleibt, überprüfen Sie, ob eine Fehlermeldung (Drucker - die Argumente sind ungültig) hinter dem Hauptinstallationsfenster angezeigt wird. Diese Meldung wird angezeigt, wenn der Druckspoolerdienst noch nicht gestartet wurde. Warten Sie, bis der Druckspoolerdienst gestartet wird oder starten Sie den Dienst manuell. Wenn Sie im Meldungsfeld auf "OK" klicken, wird die Installation fortgesetzt. [#385526]
- Für VDAs vor 7 werden die Daten der Benutzer u. U. nicht in Director angezeigt, selbst wenn Sie die Windows-Remoteverwaltung (WinRM) für diese VDAs richtig konfiguriert haben. Starten Sie den WinRM-Dienst neu, um dieses Problem zu beheben. Die Daten werden wie erwartet in Director angezeigt. [#392047]
- In sicheren Umgebungen wird ein neuer VDA möglicherweise nicht bei einem Controller registriert. Vor allem bei der Installation von einem VDA, der eine lokale Sicherheitsrichtlinieneinstellung hat, die nur Administratoren den Zugriff auf den Computer vom Netzwerk geben, wird der VDA installiert, die Registrierung beim Controller schlägt jedoch fehl. Stattdessen wird eine Warnung angezeigt, dass die Benutzerrechte nicht richtig konfiguriert sind. Weitere Informationen finden Sie unter [CTX117248](#). [#336203]
- Benutzer von Receiver für Windows können sich nicht mit Passthrough-Authentifizierung an Stores anmelden, selbst wenn die Domänen-Passthrough-Authentifizierung im StoreFront-Authentifizierungsdienst aktiviert ist. Um dieses Problem zu beheben, führen Sie den Befehl "Set-BrokerSite -TrustRequestsSentToTheXmlServicePort \$true" in einer Windows PowerShell-Eingabeaufforderung auf dem Controller aus. [#330775]
- ICA-Roundtripüberprüfungen werden nicht unterstützt, wenn der Legacygrafikmodus mit einer Richtlinie für Windows Server 2012-VDAs angegeben ist. Für dieses Problem gibt es keinen Workaround. Hinweis: Windows 8-VDAs unterstützen den Legacymodus nicht, sind daher von diesem Problem nicht betroffen. [#394824]
- Wenn Sie Maschinenerstellungsdienste (MCS) zum Bereitstellen von Maschinen nutzen, verwenden Sie nur ASCII-Zeichen für die Maschinennamen.

Durch die Verwendung von Nicht-ASCII-Zeichen werden andere nicht in Verbindung stehende Zeichen als Hostnamen in bereitgestellten Maschinen angezeigt. [#415134]

- Beim Erstellen eines Maschinenkatalogs mit Maschinenerstellungsdienste (MCS) führt die Verwendung eines Namens mit dem Akzent Gravis (`) zu einem Fehler: Vermeiden Sie die Verwendung dieses Zeichens in Maschinenkatalognamen [#414419].
- Es können leere oder schwarze Bildschirme angezeigt werden, wenn Benutzer sich bei Windows Server 2012 R2-Maschinen anmelden, und eine Verbindung mit dem Server über eine RDP-Verbindung ist nicht möglich. Außerdem können auf der Serverkonsole einige Systemprozesse, wie Task-Manager, nicht ausgeführt werden, sodass ein System Kaltstart durchgeführt werden muss.

Ändern Sie zur Vermeidung dieses Problems die Richtlinieneinstellung für "Enhanced Desktop Experience" auf "Nicht zugelassen" für Server 2012-Bereitstellungsgruppen. [#425255]

- Wenn Sie bei Windows Server 2012 R2 oder einem älteren Domänencontroller vorgeben, dass Benutzer bei der nächsten Anmeldung ihr Kennwort ändern müssen, tritt ein Fehler auf. Entfernen Sie zur Umgehung des Problems das Microsoft-Update KB2883201. [#438725]
- Wenn sich Benutzer direkt über RDP mit VDAs verbinden, meldet der Controller manchmal zusätzliche Sitzungen, die nicht vorhanden sind, und diese nicht bestehenden Sitzungen werden in Studio als "Verbunden" angezeigt, bis der VDA neu gestartet wird. [#385823]
- Wenn für Citrix StoreFront unter dem Browser Firefox oder Chrome die Passthrough-Authentifizierung implementiert ist, werden die Benutzer beim Starten von Anwendungen aufgefordert, Anmeldeinformationen einzugeben. [#441487]
- Meldet sich unter Citrix Receiver für Web mit Domänen-Passthrough-Konfiguration ein Benutzer mit Smartcard bei einem Gerät an und startet dann einen veröffentlichten Desktop, kann die Verbindung fehlschlagen. Zur Problemlösung bearbeiten Sie die ICA-Standarddatei (z. B. C:\inetpub\wwwroot\Citrix\Store\App_Data\default.ica) und fügen Sie dem Abschnitt [Application] die folgende Zeile hinzu:

DisableCtrlAltDel=False
[#452813]

XenApp 7.5 und frühere Versionen

Sep 29, 2015

Die Migration von XenApp zur FlexCast Management Architecture (FMA) zieht Änderungen an Konzepten und Terminologie nach sich. In diesem Abschnitt werden Konzepte und Terminologie bezüglich XenApp 6-Entitäten im Umfeld von XenApp 7 erläutert.

Der Einfachheit halber bezieht sich "XenApp 7" in diesem Abschnitt auf alle aktuellen Releases (z. B. XenApp 7.5) und "XenApp 6" auf alle zugehörigen Nebenreleases (z. B. XenApp 6.5).

In der folgenden Tabelle sind XenApp 6-Funktionselemente XenApp 7 zugeordnet, auch wenn sie diesen nicht exakt entsprechen:

Element in XenApp 6	Element in XenApp 7
Independent Management Architecture (IMA)	FlexCast Management Architecture (FMA)
Farm	Bereitstellungssite
Workergruppe	Sitzungsmaschinenkatalog Bereitstellungsgruppe
Worker	Virtual Delivery Agent Serverbetriebssystemmaschine Desktopbetriebssystemmaschine
Zonen- und Datensammelpunkt	Delivery Controller
Delivery Services Console	Citrix Studio und Citrix Director
Veröffentlichen von Anwendungen	Bereitstellen von Anwendungen
Datenspeicher	Datenbank
Lastauswertungsprogramm	Lastverwaltungsrichtlinie
Administrator	Delegierter Administrator Rolle Geltungsbereich

Element in XenApp 6	Element in XenApp 7
---------------------	---------------------

FlexCast Management Architecture

Die FlexCast Management Architecture (FMA) ist eine dienstorientierte Architektur, die über Citrix Technologien übergreifend Interoperabilität und Verwaltungsmodularität ermöglicht. Die FMA bietet eine Plattform für die Anwendungsbereitstellung, Mobilität, Dienste, flexible Bereitstellung und Cloudverwaltung.

Die FMA ersetzt die Independent Management Architecture (IMA) in XenApp 6.

Elemente der neuen Architektur

Bereitstellungssites

Farmen waren in XenApp 6 Objekte der obersten Ebene. In XenApp 7 ist dies die Bereitstellungs-site. Über Sites werden Benutzergruppen Anwendungen und Desktops angeboten.

Die FMA erfordert, dass Sie in einer Domäne sind, um eine Site bereitzustellen. Beispiel: Zum Installieren der Server muss Ihr Konto lokale Administratorrechte haben und in Active Directory ein Domänenbenutzer sein.

Sitzungsmaschinenkataloge und Bereitstellungsgruppen

Maschinen, auf denen Anwendungen gehostet werden, gehörten in XenApp 6 zu Workergruppen, um eine effiziente Verwaltung von Anwendungen und Serversoftware zu ermöglichen. Administratoren konnten bei der Anwendungsverwaltung und beim Lastausgleich alle Maschinen in einer Workergruppe als einzelne Einheit behandeln. Ordner wurden verwendet, um Anwendungen und Maschinen zu organisieren.

In XenApp 7 wird eine Kombination aus Sitzungsmaschinenkatalogen und Bereitstellungsgruppen zur Verwaltung von Maschinen, Lastausgleich und gehosteten Anwendungen oder Desktops verwendet.

Ein Sitzungsmaschinenkatalog ist eine Sammlung von Maschinen, die gleich konfiguriert und verwaltet werden. Eine virtuelle oder physische Maschine gehört nur zu einem Katalog. Auf allen Maschinen des Katalogs sind die gleichen Anwendungen und Desktops verfügbar.

Bereitstellungsgruppen dienen zur Bereitstellung von Anwendungen und Desktops für Benutzer. Eine Bereitstellungsgruppe kann Maschinen aus mehreren Maschinenkatalogen enthalten und ein Maschinenkatalog kann Maschinen zu mehreren Bereitstellungsgruppen beitragen. Eine einzelne Maschine kann jedoch nur zu einer Bereitstellungsgruppe gehören. Sie können die auf Maschinen ausgeführte Software über die Kataloge, zu denen die Maschinen gehören, verwalten. Der Zugriff der Benutzer auf Anwendungen wird über die Bereitstellungsgruppen verwaltet.

Virtual Delivery Agents

Der Virtual Delivery Agent (VDA) ermöglicht Verbindungen mit Anwendungen und Desktops. Der VDA wird auf der Maschine installiert, auf der die Anwendungen oder virtuellen Desktops für den Benutzer ausgeführt werden. Er ermöglicht eine Registrierung der Maschinen bei Delivery Controllern und die Verwaltung der HDX-Verbindung (High Definition eXperience) auf Benutzergeräten.

In XenApp 6 wurden auf Maschinen in Workergruppen Anwendungen für die Benutzer ausgeführt und die Maschinen kommunizierten mit Datensammelpunkten. In XenApp 7 kommuniziert der VDA mit Delivery Controllern, die die Benutzerverbindungen verwalten.

Eine Installation des VDA ist möglich auf:

- Serverbetriebssystemmaschinen: Maschinen, auf denen ein Windows Server-Betriebssystem ausgeführt wird
- Desktopbetriebssystemmaschinen: Maschinen, auf denen ein Windows-Desktopbetriebssystem ausgeführt wird

Delivery Controller

In XenApp 6 war ein Zonenmaster für Verbindungsanfragen von Benutzern und die Kommunikation mit Hypervisors zuständig. In XenApp 7 werden Verbindungsanfragen von den Controllern der Site verteilt und behandelt.

XenApp 6-Zonen ermöglichten das Aggregieren von Servern und Replizieren von Daten über WAN-Verbindungen. In XenApp 7 gibt es zwar keine exakte Entsprechung für Zonen, dennoch können Sie Benutzern Anwendungen bereitstellen, die WANS

und Standorte überqueren. Sie können Bereitstellungssites für bestimmte geografische Standorte oder Datacenter erstellen und dann den Benutzern Zugriff auf mehrere Bereitstellungssites gewähren. App Orchestration mit XenApp 7 enthält Funktionen zum Verwalten mehrerer Sites in mehreren geografischen Bereichen.

Citrix Studio und Citrix Director

Mit der Studio-Konsole können Sie Umgebungen konfigurieren und Benutzern Zugriff auf Anwendungen und Desktops gewähren. Studio ersetzt die Delivery Services Console in XenApp 6.

Administratoren verwenden Director, um die Umgebung zu überwachen, Benutzergeräte zu spiegeln und IT-Probleme zu behandeln.

Bereitstellen von Anwendungen

In XenApp 6 wurden mit dem Assistenten zur Anwendungsveröffentlichung Anwendungen vorbereitet und für Benutzer bereitgestellt. Bei XenApp 7 dient Studio zum Erstellen und Hinzufügen von Anwendungen für Benutzer, die zu einer Bereitstellungsgruppe gehören. Mit Studio konfigurieren Sie zunächst eine Site, erstellen dann Maschinenkataloge und dann die Bereitstellungsgruppen in diesen Katalogen. Mit Bereitstellungsgruppen wird festgelegt, welche Benutzer Zugriff auf die bereitgestellten Anwendungen haben.

Datenbank

In XenApp 7 wird der IMA-Datenspeicher nicht für Konfigurationsinformationen verwendet. Stattdessen wird eine Microsoft SQL Server-Datenbank zum Speichern der Konfigurations- und Sitzungsinformationen verwendet.

Lastverwaltungsrichtlinie

In XenApp 6 verwendeten Lastauswertungsprogramme vordefinierte Messungen zur Bestimmung der Last auf einer Maschine. Benutzerverbindungen konnten weniger ausgelasteten Maschinen zugeordnet werden.

In XenApp 7 erfolgt der Lastausgleich durch Lastverwaltungsrichtlinien.

Delegierte Administratoren

In XenApp 6 wurden benutzerdefinierte Administratoren erstellt und ihnen wurden Berechtigungen auf der Basis von Ordnern und Objekten zugewiesen. In XenApp 7 basieren benutzerdefinierte Administratoren auf Rollen-/Geltungsbereichspaaren. Eine Rolle repräsentiert eine Stellenfunktion. Ihr sind definierte Berechtigungen zugewiesen, die eine Delegierung erlauben. Ein Geltungsbereich steht für eine Sammlung von Objekten. Vordefinierte Administratorrollen haben spezifische Berechtigungssätze, z. B. für Helpdesk, Anwendungen, Hosting und Kataloge. Beispiel: Helpdeskadministratoren können nur mit einzelnen Benutzern in bestimmten Sites arbeiten, während Volladministratoren die gesamte Bereitstellung überwachen und systemweite IT-Probleme behandeln können.

Systemanforderungen

Dec 07, 2015

In diesem Abschnitt werden die zum Zeitpunkt dieser Veröffentlichung unterstützten Systeme für den Delivery Controller, Citrix Studio, Citrix Director und Virtual Delivery Agents aufgeführt. Systemanforderungen für andere Komponenten (z. B. StoreFront, Hostsysteme, Receiver und Plug-Ins sowie Provisioning Services) finden Sie in der zugehörigen Dokumentation. Informationen zu den Anforderungen von HDX und Desktop Lock finden Sie unter [HDX](#) und [Desktop Lock](#).

Sofern nicht anders angegeben, wird erforderliche Software (z. B. .NET und C++-Pakete), wenn sie nicht auf einer Maschine erkannt werden, automatisch bereitgestellt. Das Citrix Installationsmedium enthält außerdem einige erforderliche Softwarekomponenten.

Das Installationsmedium enthält mehrere Komponenten von Drittanbietern. Bevor Sie diese Citrix Software verwenden, überprüfen Sie, ob Sicherheitsupdates von Drittanbietern nötig sind und installieren Sie sie.

Die Speicherplatzangaben für die Komponenten sind Schätzwerte und verstehen sich zuzüglich des für Produktimage, Betriebssystem und anderer Software erforderlichen Speicherplatzes.

Wenn Sie die Kernkomponenten (Controller, Studio, Director, StoreFront und Lizenzierung) auf einem Server installieren, benötigen Sie für die Evaluierung des Produkts mindestens 3 GB Arbeitsspeicher; mehr Speicher wird empfohlen, wenn Sie eine Umgebung für Benutzer ausführen. Die Leistung schwankt abhängig von der genauen Konfiguration, einschließlich der Anzahl der Benutzer, Anwendungen und Desktops sowie anderen Faktoren.

Lesen Sie auf jeden Fall vor Beginn der Installation den Abschnitt [Vorbereiten der Installation](#).

Delivery Controller

Unterstützte Betriebssysteme:

- Windows Server 2012 R2, Standard und Datacenter Edition
- Windows Server 2012, Standard und Datacenter Edition
- Windows Server 2008 R2 SP1, Standard, Enterprise und Datacenter Edition

Anforderungen:

- Speicherplatz auf der Datenträger: 100 MB
- Microsoft .NET Framework 3.5.1 (nur Windows Server 2008 R2)
- Microsoft .NET Framework 4.5
- Windows PowerShell 2.0 (in Windows Server 2008 R2 enthalten) oder 3.0 (in Windows Server 2012 R2 und Windows Server 2012 enthalten)
- Visual C++ 2005, 2008 SP1 und 2010 Redistributable-Pakete

Datenbank

Unterstützte Versionen von Microsoft SQL Server für die Sitekonfigurationsdatenbank (die zunächst die Konfigurationsprotokollierungsdatenbank und die Überwachungsdatenbank enthält):

- SQL Server 2012 SP1, Express, Standard und Enterprise Edition Standardmäßig wird SQL Server 2012 Express zusammen mit dem Controller installiert, wenn keine vorhandene unterstützte SQL Server-Installation erkannt wird.
- SQL Server 2008 R2 SP2, Express, Standard, Enterprise und Datacenter Edition

Die folgenden Datenbankfeatures werden unterstützt (außer bei SQL Server Express, das nur den eigenständigen Modus unterstützt):

- SQL Server-Cluster-Instanzen
- SQL Server-Spiegelung
- SQL Server 2012-AlwaysOn-Verfügbarkeitsgruppen

Die Windows-Authentifizierung ist für Verbindungen zwischen dem Controller und der SQL Server-Datenbank erforderlich.

Weitere Informationen zu den aktuell unterstützten Versionen der Datenbanken finden Sie unter <http://support.citrix.com/article/CTX114501>.

Studio

Unterstützte Betriebssysteme:

- Windows 8.1, Professional und Enterprise Edition
- Windows 8, Professional und Enterprise Edition
- Windows 7, Professional, Enterprise und Ultimate Edition
- Windows Server 2012 R2, Standard und Datacenter Edition
- Windows Server 2012, Standard und Datacenter Edition
- Windows Server 2008 R2 SP1, Standard, Enterprise und Datacenter Edition

Anforderungen:

- Speicherplatz auf der Datenträger: 75 MB
- Microsoft .NET Framework 3.5 SP1 (nur Windows Server 2008 R2 und Windows 7)
- Microsoft .NET Framework 4.5
- Microsoft Management Console 3.0 (in allen unterstützten Betriebssystemen enthalten)
- Windows PowerShell 2.0 (in Windows 7 und Windows Server 2008 R2 enthalten) oder 3.0 (in Windows 8.1, Windows 8, Windows Server 2012 R2 und Windows Server 2012 enthalten)

Director

Unterstützte Betriebssysteme:

- Windows Server 2012 R2, Standard und Datacenter Edition
- Windows Server 2012, Standard und Datacenter Edition
- Windows Server 2008 R2 SP1, Standard, Enterprise und Datacenter Edition

Anforderungen:

- Speicherplatz auf dem Datenträger: 50 MB
- Microsoft .NET Framework 4.5
- Microsoft Internetinformationsdienste (IIS) 7.0 und ASP.NET 2.0. Wenn diese Komponenten nicht auf Ihrem Server installiert sind, werden Sie möglicherweise aufgefordert, das Windows Server-Installationsmedium einzulegen. Sie werden dann installiert.
- Unterstützte Browser zum Anzeigen von Director:
 - Internet Explorer 11, 10 und 9.
Der Kompatibilitätsmodus wird für Internet Explorer nicht unterstützt. Für den Zugriff auf Director müssen Sie die empfohlenen Webbrowsereinstellungen verwenden. Akzeptieren Sie bei der Installation von Internet Explorer die Standardeinstellung zur Verwendung der empfohlenen Sicherheits- und Kompatibilitätseinstellungen. Wenn der Browser bereits ohne die empfohlenen Einstellungen installiert ist, navigieren Sie auf "Extras > Internetoptionen > Erweitert" > Zurücksetzen" und folgen Sie den Anweisungen.
 - Firefox ESR (Extended Support Release)
 - Chrome

Virtual Delivery Agent (VDA) für Windows-Desktopbetriebssysteme

Unterstützte Betriebssysteme:

- Windows 8.1, Professional und Enterprise Edition
- Windows 8, Professional und Enterprise Edition
- Windows 7 SP1, Professional, Enterprise und Ultimate Edition

Anforderungen:

- (Nur für Windows 7 SP1-Systeme.) Microsoft .NET Framework 3.5 1: Bei Installation über die Befehlszeile müssen Sie dieses manuell installieren, bevor Sie den VDA installieren.
- (Nur für Windows 7 SP1-Systeme; andere unterstützte Betriebssysteme enthalten standardmäßig bereits mindestens .NET Framework 4.) Microsoft .NET Framework 4.0
- Microsoft Visual C++ 2005, 2008 und 2010 Runtime (32-Bit oder 64-Bit, je nach Plattform)

Remote-PC-Zugriff verwendet diesen VDA, den Sie auf physischen Büro-PCs installieren.

Mehrere Multimediabeschleunigungsfunktionen (z. B. HDX MediaStream-Windows Media-Umleitung) erfordern, dass Microsoft Media Foundation auf dem Computer installiert wird, auf dem der VDA installiert ist. Wenn Media Foundation nicht installiert ist, werden die Multimediabeschleunigungsfeatures nicht installiert und sind nicht funktionsfähig. Entfernen Sie Media Foundation nicht nach der Installation der Citrix Software von der Maschine, sonst können sich Benutzer nicht an der Maschine anmelden. Bei den meisten Editionen von Windows 8.1, Windows 8 und Windows 7 ist Media Foundation bereits installiert und kann nicht entfernt werden. Bei N-Editionen sind bestimmte medienrelevante Technologien nicht enthalten; Sie können die Software von Microsoft oder einem Drittanbieter beziehen.

Sie können einen VDA der Version 7.5 nicht auf einer Maschine unter Windows XP oder Windows Vista installieren. Sie können bei diesen Betriebssystemen jedoch bei Bedarf eine Vorversion von Virtual Desktop Agent installieren. Weitere Informationen finden Sie in den Anweisungen zum Installieren einer früheren Version von VDA auf diesen Betriebssystemen. Die Version von Remote-PC-Zugriff in diesem Release wird nicht auf Windows Vista-Betriebssystemen unterstützt.

Virtual Delivery Agent (VDA) für Windows-Serverbetriebssysteme

Unterstützte Betriebssysteme:

- Windows Server 2012 R2, Standard und Datacenter Edition
- Windows Server 2012, Standard und Datacenter Edition
- Windows Server 2008 R2 SP1, Standard, Enterprise und Datacenter Edition

Das Installationsprogramm stellt die folgenden Anforderungen automatisch bereit, die auch auf den Citrix Installationsmedien in den Ordnern \Support zur Verfügung stehen:

- (Nur für Windows 2008 R2-Systeme.) Microsoft .NET Framework 3.5 1: Bei Installation über die Befehlszeile müssen Sie dieses manuell installieren, bevor Sie den VDA installieren.
- (Nur für Windows 2008 R2 SP1-Systeme; andere unterstützte Betriebssysteme enthalten standardmäßig bereits mindestens .NET Framework 4.) Microsoft .NET Framework 4.5.1
- Microsoft Visual C++ 2005, 2008 und 2010 Runtime (32 Bit und 64 Bit)

Das Installationsprogramm installiert und aktiviert automatisch die Rollendienste für Remotedesktopdienste, wenn sie nicht bereits installiert und aktiviert sind.

Mehrere Multimediabeschleunigungsfunktionen (z. B. HDX MediaStream-Windows Media-Umleitung) erfordern, dass Microsoft Media Foundation auf dem Computer installiert wird, auf dem der VDA installiert ist. Wenn Media Foundation nicht installiert ist, werden die Multimediabeschleunigungsfeatures nicht installiert und sind nicht funktionsfähig. Entfernen

Sie Media Foundation nicht nach der Installation der Citrix Software von der Maschine, sonst können sich Benutzer nicht an der Maschine anmelden. Bei den meisten Editionen von Windows Server 2012 R2, Windows Server 2012 und Windows Server 2008 R2 wird Media Foundation über den Server-Manager installiert (bei Windows Server 2012 R2 und Windows Server 2012: ServerMediaFoundation, bei Windows Server 2008 R2: DesktopExperience). Bei N-Editionen sind bestimmte medienrelevante Technologien nicht enthalten; Sie können die Software von Microsoft oder einem Drittanbieter beziehen.

Der Druckspoolerdienst ist auf dem Windows-Server standardmäßig aktiviert. Wenn Sie diesen Dienst deaktivieren, können Sie keinen VDA für Windows-Serverbetriebssysteme installieren. Daher müssen Sie sicherstellen, dass der Dienst aktiviert ist, bevor ein VDA installiert wird.

Virtualisierungsressourcen

Unterstützte Plattformen:

- XenServer:
 - XenServer 6.5
 - XenServer 6.2 SP1 und Hotfixes (Sie müssen SP1 anwenden, damit die Anwendung zukünftiger Hotfixes möglich ist)
 - XenServer 6.1
 - XenServer 6.0.2
- VMware vSphere: Der "Linked Mode"-Betrieb von vSphere vCenter wird nicht unterstützt.
 - 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
- System Center Virtual Machine Manager: Enthält alle Versionen von Hyper-V, die mit den unterstützten Versionen von System Center Virtual Machine Manager registriert werden können.
 - System Center Virtual Machine Manager 2012 R2
 - System Center Virtual Machine Manager 2012 SP1
 - System Center Virtual Machine Manager 2012

Sie können dieses Produkt auch in den folgenden Cloudumgebungen bereitstellen:

- Amazon Web Services (AWS)
 - Sie können Anwendungen und Desktops auf unterstützten Windows Server-Betriebssystemen bereitstellen. AWS bietet keine Windows Server 2012 R2-Instanzen.
 - SQL Server 2012 Enterprise ist für AWS nicht verfügbar.
 - AWS bietet keine Desktopbetriebssysteminstanzen.
 - Amazon Relational Database Service (RDS) wird nicht unterstützt.
 - Weitere Informationen finden Sie in der AWS-Dokumentation und unter [CTX140427](#).
- Citrix CloudPlatform
 - Mindestens unterstützte Version ist 4.2.1 mit Hotfixes 4.2.1-4
 - Bereitstellungen wurden mit XenServer 6.2 (mit Service Pack 1 und Hotfix XS62ESP1003) und vSphere 5.1-Hypervisors getestet.
 - CloudPlatform bietet keine Unterstützung für VMware vSphere 5.5 und Hyper-V-Hypervisors.
 - Weitere Informationen zu Systemanforderungen für Linux-basierte Systeme finden Sie in der CloudPlatform-Dokumentation und unter [CTX140428](#).

Weitere Informationen:

- Informationen zu Systemanforderungen und zur Installation finden Sie in der Dokumentation zu der Plattform bzw. Cloudumgebung.
- Aktualisierte Informationen zur Hypervisorunterstützung finden Sie unter [CTX131239](#).

Die folgenden Kombinationen aus Virtualisierungsressourcen- und Speichertechnologien werden für Maschinenerstellungsdienste und für das Hinzufügen von Active Directory-Konten in VMs zur Laufzeit unterstützt: Mit einem Sternchen (*) gekennzeichnete Kombinationen werden empfohlen.

Virtualisierungsressource	Lokale Datenträger	NFS	Blockspeicher	Speicherlink
XenServer	Ja	Ja *	Ja	Nein
VMware	Ja (nicht vMotion oder dynamische Platzierung)	Ja *	Ja	Nein
Hyper-V	Ja	Nein	Ja * (erfordert freigegebene Clustervolumes)	Nein

Das Wake-On-LAN-Feature von Remote-PC-Zugriff erfordert Microsoft System Center Configuration Manager 2012. Siehe [Microsoft System Center Configuration Manager und Remote-PC-Zugriff-Wake-On-LAN](#).

Sonstiges

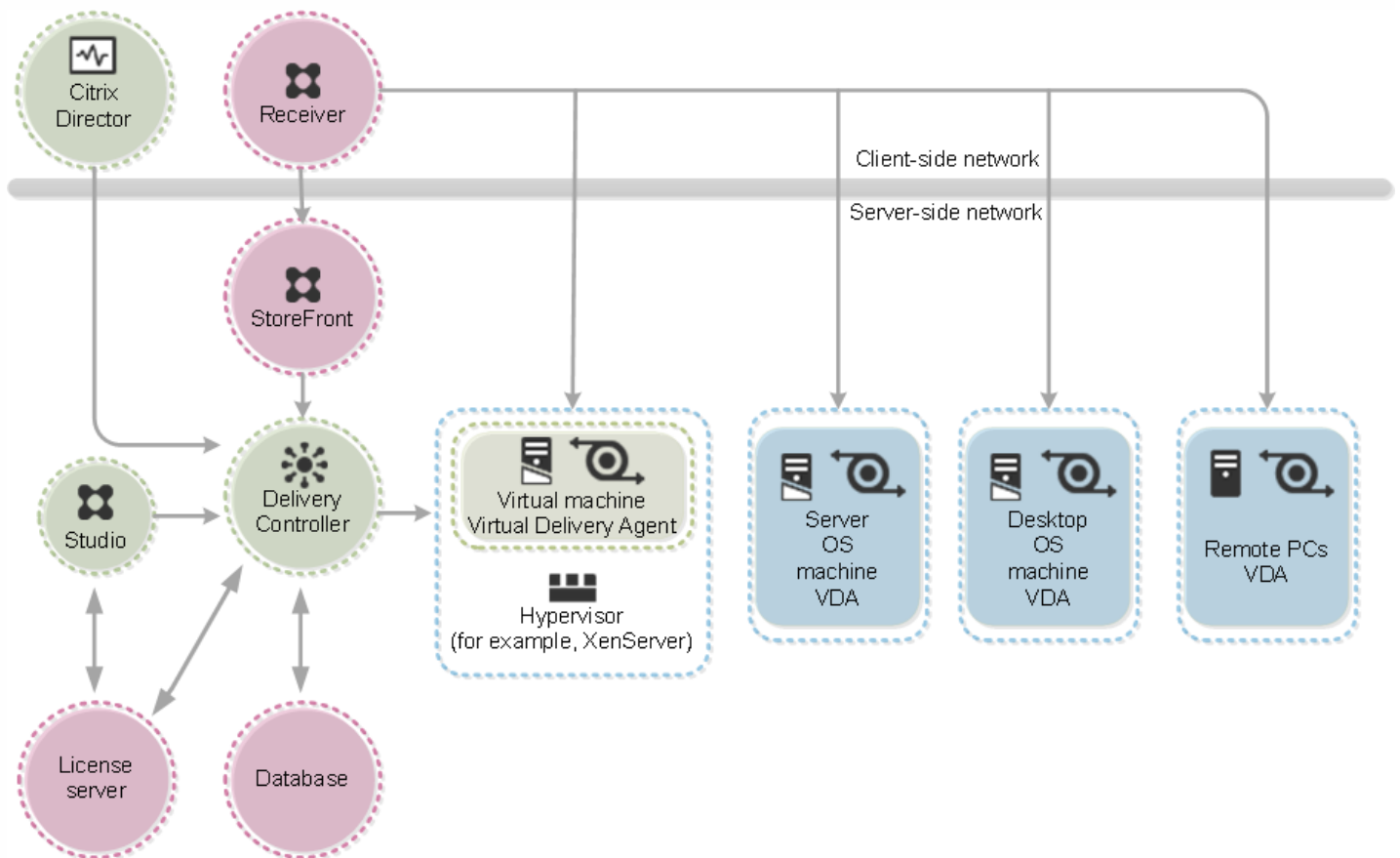
- Citrix empfiehlt die Installation der Komponentensoftwareversionen auf dem Installationsmedium für dieses Release.
 - StoreFront erfordert 2 GB Arbeitsspeicher. Informationen zu Systemanforderungen finden Sie in der Dokumentation von StoreFront. StoreFront 2.0 ist die in diesem Release unterstützte Mindestversion.
 - Wenn Sie Provisioning Services mit diesem Release verwenden, wird als Mindestversion Provisioning Services 7.0 unterstützt.
 - Citrix Lizenzserver erfordert 40 MB Speicherplatz auf dem Datenträger. Informationen zu den Systemanforderungen finden Sie in der Dokumentation der Lizenzierung. Es wird nur Citrix Lizenzserver für Windows unterstützt. Die unterstützte Mindestversion ist 11.11.
- Universeller Druckserver - Der universelle Druckserver umfasst Client- und Serverkomponenten. Die UPClient-Komponente ist in der VDA-Installation enthalten. Die UPServer-Komponente (die Sie auf jedem Druckserver installieren, auf dem freigegebene Drucker sind, die Sie mit dem universellen Druckertreiber von Citrix in Benutzersitzungen bereitstellen möchten) wird unter folgenden Systemen unterstützt:
 - Windows Server 2008 R2 SP1
 - Windows Server 2008 (32 Bit)
- Die Microsoft-Gruppenrichtlinien-Verwaltungskonsolle ist erforderlich, wenn Sie Citrix Richtlinieninformationen in Active Directory und nicht in der Sitekonfigurationsdatenbank speichern. Weitere Informationen finden Sie in der Dokumentation von Microsoft.
- Standardmäßig wird zusammen mit einem VDA Receiver für Windows installiert. Informationen zu den Systemanforderungen für andere Plattformen finden Sie in der Dokumentation von Receiver für Windows.
- Receiver für Linux und Receiver für Mac stehen auf dem Produktinstallationsmedium zur Verfügung. Weitere Informationen zu Systemanforderungen finden Sie in der jeweiligen Dokumentation.
- Wenn Sie Access Gateway-Versionen vor 10.0 mit diesem Release verwenden, werden Windows 8.1- und Windows 8-Clients nicht unterstützt.

Grundlegende Konzepte

Sep 29, 2015

Wenn Sie bereits mit XenDesktop- oder XenApp-Umgebungen vertraut sind, lohnt es sich, die neuen Konzepte und Komponenten in diesem Release kennenzulernen und zu erfahren, wie diese funktionieren und miteinander kommunizieren. In der neuen Architektur wurden XenDesktop und XenApp samt Verwaltungs- und Bereitstellungskomponenten zusammengeführt, um Administratoren ein einheitliches Verwaltungserlebnis zu geben.

Diese Abbildung zeigt die Hauptkomponenten in einer typischen Bereitstellung.



Abgebildete Kernkomponenten in diesem Release:

Director: Ein webbasiertes Tool, mit dem die Support- und Helpdesk-Teams eine XenDesktop-Umgebung überwachen, potenziell systembedrohende Probleme rechtzeitig behandeln und Unterstützung für Endbenutzer leisten können. Zudem können Sie über Microsoft Remoteunterstützung auch Benutzersitzungen anzeigen und steuern.

Receiver: Citrix Receiver wird auf Benutzergeräten installiert und bietet Benutzern schnellen, sicheren Self-Service-Zugriff auf Dokumente, Anwendungen und Desktops von allen Geräten des Benutzers, u. a. Smartphones, Tablets und PCs. Receiver bietet bedarfsgesteuerten Zugriff auf Windows-, Web- und SaaS-Anwendungen.

StoreFront: Hiermit werden Benutzer an Sites authentifiziert, die Ressourcen hosten und die Anwendungen und Desktops verwalten, auf die Benutzer zugreifen.

Studio: Dies ist die Verwaltungskonsolle, mit der Sie die Bereitstellung konfigurieren und verwalten, wobei keine separaten Verwaltungskonsolen für die Bereitstellung von Anwendungen und Desktops benötigt werden. Studio bietet zahlreiche

Assistenten, die Ihnen bei der Einrichtung Ihrer Umgebung, dem Erstellen der Arbeitslast zum Hosten von Anwendungen und Desktops sowie beim Zuweisen von Anwendungen und Desktops zu Benutzern behilflich sind.

Lizenzserver: Der Lizenzserver verwaltet die Produktlizenzen. Sie müssen mindestens einen Lizenzserver zum Speichern und Verwalten Ihrer Lizenzdateien erstellen.

Delivery Controller: Wird auf Servern im Datacenter installiert. Der Delivery Controller besteht aus Diensten, die mit dem Hypervisor kommunizieren, Anwendungen und Desktops verteilen, den Benutzerzugriff authentifizieren und verwalten und Verbindungen zwischen Benutzern und deren virtuellen Desktops und Anwendungen vermitteln. Der Controller verwaltet den Zustand der Desktops, startet und hält sie basierend auf dem Bedarf und der administrativen Konfiguration an. In bestimmten Editionen ermöglicht der Controller die Installation der Profilverwaltung, mit der Sie personalisierte Einstellungen in virtualisierten oder physischen Windows-Umgebungen verwalten. Jede Site hat einen oder mehrere Delivery Controller.

XenServer: XenServer ist eine Unternehmenslösung für VM-Infrastrukturen, die die Grundlage für die Bereitstellung virtueller Desktops erstellt und erweiterte Verwaltungsfunktionen zur Verfügung stellt. Unter XenServer können mehrere VMs ausgeführt werden, wobei die erweiterten Virtualisierungsfunktionen der aktuellen virtualisierungsfähigen Prozessoren von Intel und AMD genutzt werden. Weitere Informationen zu XenServer finden Sie in der XenServer-Dokumentation in den eDocs.

Virtual Delivery Agent (VDA): VDA wird auf Server- oder Arbeitsstationsbetriebssystemen installiert und ermöglicht Verbindungen für Desktops und Apps. Installieren Sie für Remote-PC-Zugriff den VDA auf dem Büro-PC.

Maschinenerstellungsdienste: Eine Sammlung von Diensten, die zusammenarbeiten und bei Bedarf virtuelle Server und Desktops von einem Masterdesktopimage erstellen, die Speichernutzung optimieren und den Benutzern bei jeder Anmeldung eine einwandfreie virtuelle Maschine bereitstellen. Die Maschinenerstellungsdienste sind vollständig in Citrix Studio integriert und werden dort verwaltet.

Windows-Serverbetriebssystemmaschinen: VMs oder physische Maschinen, die auf dem Windows-Serverbetriebssystem basieren und für die Bereitstellung von Anwendungen oder gehosteten, freigegebenen Desktops für Benutzer verwendet werden.

Desktopbetriebssystemmaschinen: VMs oder physische Maschinen, die auf dem Windows-Desktopbetriebssystem basieren und den Benutzern persönliche Desktops oder Anwendungen von Desktopbetriebssystemen bereitstellen.

Remote-PC-Zugriff: Wenn Benutzergeräte auf einer Positivliste enthalten sind, können die Benutzer von einem beliebigen Gerät aus, auf dem Citrix Receiver ausgeführt wird, auf ihre Büro-PCs zugreifen.

Zusätzliche Komponenten bieten die folgenden Features:

Sichere Bereitstellung: Wenn Benutzer eine Verbindung von außerhalb der Unternehmensfirewall herstellen, können diese Verbindungen in diesem Release mit Citrix NetScaler Gateway (früher Access Gateway) mit SSL gesichert werden. NetScaler Gateway oder NetScaler VPX virtuelles Gerät ist ein SSL-VPN-Gerät, das in der DMZ (demilitarisierten Zone) bereitgestellt wird und sicheren, zentralen Zugriff über die Unternehmensfirewall bietet.

WAN-Optimierung: In Bereitstellungen, in denen virtuelle Desktops den Benutzern an Remotestandorten bereitgestellt werden, wie z. B. Zweigstellen, kann mit Citrix CloudBridge (früher Citrix Branch Repeater oder WANScaler) die Leistung optimiert werden. Repeater erhöhen die Leistung in WAN-Netzwerken. Mit Repeatern im Netzwerk erleben Benutzer in Zweigstellen eine LAN-ähnliche Leistung über das WAN. CloudBridge kann bestimmten Teilen der Benutzererfahrung Priorität geben, damit sich beispielsweise die Benutzererfahrung in der Zweigstelle nicht verschlechtert, wenn eine große Datei oder ein großer Druckauftrag über das Netz gesendet wird. HDX WAN-Optimierung mit CloudBridge bietet

Komprimierung mit Token sowie Entfernung von Datenduplikaten, wodurch die Bandbreitenanforderungen drastisch reduziert werden und die Leistung verbessert wird. Weitere Informationen finden Sie in der Dokumentation für Citrix CloudBridge.

Grundlegende Konzepte

Masterimage

Ein Masterimage ist ein Image, mit dem Provisioning virtuelle Maschinen (VMs) für die Benutzer erstellt. Abhängig von der verwendeten Provisioningtechnologie, kann mit dem Masterimage auch eine Maschine erstellt werden, die Anwendungen und Desktops hostet. Das auf dem Hypervisor erstellte und gespeicherte Masterimage enthält das Betriebssystem und häufig verwendete Anwendungen und Einstellungen, die Sie den Benutzern bereitstellen, z. B.:

- Antivirensoftware
- Citrix Plug-Ins
- Andere Standardprogramme

Mit einem Masterimage fangen alle Benutzer mit Desktops an, die vom Masterimage erstellt wurden. Abhängig vom Typ des Maschinenkatalogs sind Benutzeranpassungen und Systemupdates, die für den Desktop durchgeführt wurden, bei der Benutzerabmeldung persistent oder sie werden verworfen.

Anwendungen und Desktops auf dem Masterimage werden sicher verwaltet, gehostet und auf Maschinen im Datenzentrum ausgeführt, womit eine kosteneffektivere Lösung für die Anwendungsbereitstellung bereitgestellt wird.

Provisioningmethoden

Dieses Release unterstützt diese Bereitstellungsmethoden:

- **Maschinenerstellungsdienste (MCS):** Bei dieser Methode werden mit einem Masterimage in der Umgebung virtuelle Maschinen verwaltet, sodass Sie Zielgeräte über ein Masterimage verwalten und aktualisieren können. Die Maschinenerstellungsdienste sind vollständig in Citrix Studio integriert und werden dort verwaltet.
- **Provisioning Services:** Bei dieser Methode werden Computer in Echtzeit von einem freigegebenen Datenträgerimage bereitgestellt und neu bereitgestellt. Zielgeräte werden von Provisioning Services als eine Gerätesammlung verwaltet. Die Desktops und Anwendungen werden von einer Provisioning Services-vDisk bereitgestellt, deren Image von einem Masterzielgerät erstellt wurde. So können Sie die Verarbeitungsleistung der physischen Hardware oder der virtuellen Maschinen nutzen.
Provisioning Services werden in einer eigenen Konsole verwaltet.
- **Vorhandene Images:** Bei dieser Methode werden Desktops und Anwendungen, die Sie bereits auf virtuelle Maschinen im Datacenter migriert haben, verwaltet und bereitgestellt. Sie müssen die Zielgeräte individuell oder kollektiv mit ESD-Tools (Electronic Software Distribution) verwalten.

Sie wählen die Bereitstellungsmethode beim Erstellen eines Maschinenkatalogs.

Maschinenkataloge

Ein Maschinenkatalog ist eine Sammlung virtueller Maschinen und physischer Maschinen, die als eine Entität verwaltet werden. Maschinenkatalog geben Folgendes an:

- Die virtuellen oder physischen Maschinen, die für das Hosten von Anwendungen oder Desktops zur Verfügung stehen.

- Die Active Directory-Computerkonten, die diesen virtuellen Maschinen oder Computern zugewiesen sind.
- In einigen Fällen das Masterimage, das zum Erstellen der virtuellen Maschinen kopiert wird.

Es gibt zwei Kataloge:

- **Windows-Serverbetriebssystemmaschinenkatalog:** Virtuelle oder physische Maschinen, die auf dem Windows-Serverbetriebssystem basieren und Anwendungen oder gehostete, freigegebene Desktops für Benutzer bereitstellen.
- **Desktopbetriebssystemmaschinenkatalog:** Virtuelle oder physische Maschinen, die auf dem Windows-Desktopbetriebssystem basieren und den Benutzern persönliche Desktops oder Anwendungen von Desktopbetriebssystemen bereitstellen.

Bereitstellungsgruppen

Maschinen in Maschinenkatalogen sind in Bereitstellungsgruppen organisiert. Bereitstellungsgruppen stellen dieselben Anwendungen oder Desktops für Gruppen von Benutzern bereit.

Bereitstellungsgruppen ermöglichen Folgendes:

- Zuweisen von Desktops oder Anwendungen zu bestimmten Teams, Abteilungen oder Typen von Benutzern
- Verwenden von Maschinen aus mehreren Maschinenkatalogen
- Zuweisen eines Benutzers zu mehreren Maschinen
- Zuweisen mehrerer Benutzer zu einer Maschine

Serverbetriebssystemmaschinen unterstützen Folgendes:

- Desktop- und Anwendungsbereitstellungsgruppen, die sowohl Desktops als auch Anwendungen hosten.
- Anwendungsbereitstellungsgruppen, die nur Anwendungen hosten

Desktopbetriebssystemmaschinen unterstützen Folgendes:

- Identisch wie Serverbetriebssystemmaschinen

Planen

Sep 29, 2015

Mit diesem Release können Sie Ihre Bereitstellung für Ihre Organisation individuell passend erweitern. Beginnen Sie beispielsweise mit einer einfachen Standardkonfiguration, die Sie zu einem späteren Zeitpunkt für weitere Benutzergruppen bereitstellen können. Die Benutzeranforderungen sind ein entscheidender Faktor für die Bereitstellung, Ihre Testumgebung sollte also an die Anforderungen der Benutzer angepasst sein, die unmittelbare Vorteile daraus ziehen.

Falls Sie noch nicht bereit sind, den Benutzern virtuelle Desktops anzubieten, können Sie mit einer Remote-PC-Zugriff-Bereitstellung einen Anfang machen. So können Benutzer auf ihre Büro-PCs zugreifen und von Citrix HDX-Features profitieren. Zu einem späteren Zeitpunkt können Sie konventionelle virtuelle Desktops und Anwendungsbereitstellungen hinzufügen.

Grundlegende Elemente

Delivery Controller, Studio, Director, Lizenzserver und StoreFront können auf demselben Server oder auf verschiedenen Servern installiert werden. Um beispielsweise eine Bereitstellung remote zu verwalten, können Sie Studio auf einer anderen Maschine installieren als dem Server, auf dem der Controller installiert wurde.

- Controller
- Studio
- Director
- Lizenzserver: Weitere Informationen zur Lizenzierung finden Sie unter [Lizenzierung](#).
- StoreFront
- Datenbank: Standardmäßig wird bei der Siteerstellung von Studio eine Datenbank auf dem Server erstellt, auf dem Sie den Controller installieren. Diese Datenbank enthält Folgendes:
 - Sitedatenbank
 - Konfigurationsprotokollierungsdatenbank
 - Überwachungsdatenbank

Citrix empfiehlt, den Speicherort der Datenbanken für Konfigurationsprotokollierung und Überwachung nach dem Erstellen einer Site zu ändern.

Wichtig: Sie können auch eine Datenbank auf einem separaten Server verwenden. Wenn Sie eine manuell und nicht mit Studio erstellte externe Datenbank verwenden möchten, stellen Sie sicher, dass der Datenbankadministrator beim Erstellen der Datenbank die folgende Sortiereinstellung verwendet: Latin1_General_100_CI_AS_KS (wobei Latin1_General je nach Land variiert, z. B. Japanese_100_CI_AS_KS). Wenn diese Sortiereinstellung bei der Datenbankerstellung nicht angegeben wird, schlägt die anschließende Erstellung von Dienstschemas in der Datenbank fehl und es wird folgender Fehler (oder ein ähnlicher) angezeigt: ": Schema erfordert eine Datenbank, bei der Groß-/Kleinschreibung nicht beachtet wird." ist der Name des Diensts, dessen Schema erstellt wird.

- Domänencontroller mit Active Directory. Active Directory ist erforderlich. Installieren Sie XenApp, XenDesktop und die SQL Server-Datenbank nicht auf einem Domänencontroller.
Berücksichtigen Sie die möglichen Auswirkungen des gewählten Domänennamens. Vermeiden Sie Domänennamen, die möglicherweise anderswo verwendet werden. Beispielsweise wird die Zeichenfolge "client" auch für den Zugriff auf die Clientlaufwerkzuordnung verwendet.
- VMs oder physische Computer, die Desktops hosten, die Sie den Benutzern bereitstellen möchten. Sie installieren den Virtual Desktop Agent auf diesen Maschinen, um die Kommunikation und Browserverbindungen zu verwalten.
- Benutzergeräte, auf denen der entsprechende Client ausgeführt wird, damit Ihre Benutzer auf Desktops zugreifen

können.

Beispielbereitstellungen

Zwei Beispiele typischer Bereitstellungen:

- Konfiguration mit verteilten Komponenten
- Konfiguration mit mehreren Sites

Konfiguration mit verteilten Komponenten

Sie können die Komponenten Ihrer Bereitstellung auf mehrere Server verteilen oder Skalierbarkeit und Failover verbessern, indem Sie die Anzahl der Controller pro Site erhöhen. Sie können die Verwaltungskonsolen auf eigenständigen Computern installieren, damit Sie Ihre Bereitstellung remote verwalten können. Eine verteilte Bereitstellung ist für eine Infrastruktur erforderlich, die auf Remotezugriff über NetScaler Gateway (früher Access Gateway) basiert.

Abbildung 1 Konfiguration mit verteilten Komponenten



Weitere Informationen zu Citrix NetScaler Gateway für sicheren Remotezugriff finden Sie in der entsprechenden Dokumentation.

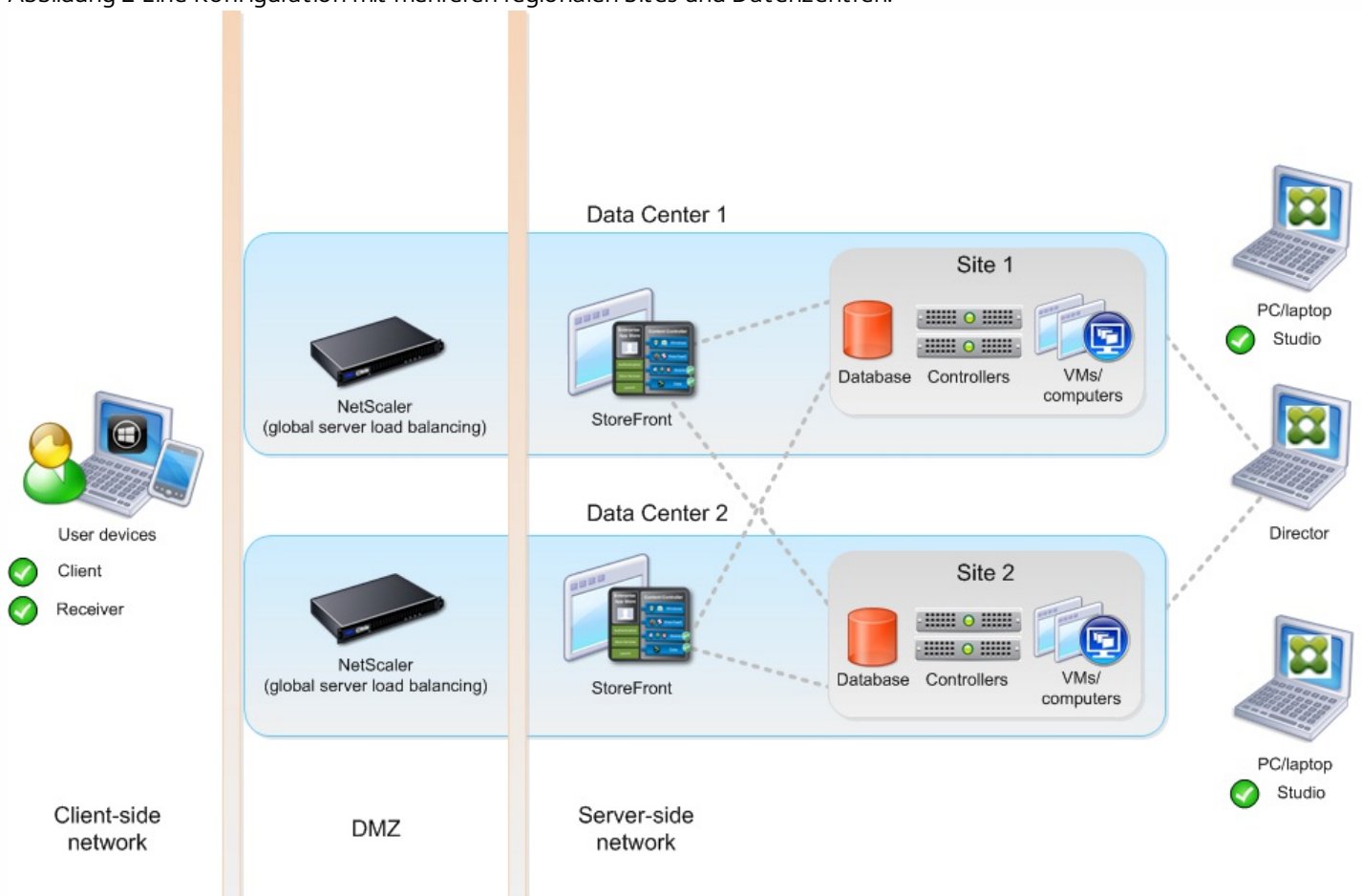
Konfiguration mit mehreren Sites

Wenn Sie mehrere regionale Sites haben, z. B. eine in Europa und eine in den USA, können Sie mit Citrix NetScaler Benutzerverbindungen zur passenden Site leiten und mit StoreFront Benutzern Desktops und Anwendungen bereitstellen.

Im folgenden Beispiel wurde eine Site in zwei Datacenter erstellt. Dadurch dass global zwei Sites statt nur einer vorhanden sind, wird unnötiger WAN-Verkehr minimiert. Mit StoreFront können Sie Ressourcen von mehreren Sites aggregieren und dann Benutzern mit NetScaler einen zentralen Zugriffspunkt bieten. Citrix NetScaler beschleunigt die Anwendungsleistung, gleicht Serverlast aus, erhöht die Sicherheit und optimiert die Benutzererfahrung. In diesem Beispiel stellen zwei NetScaler eine Konfiguration mit hoher Verfügbarkeit bereit. Die NetScaler sind für einen Lastausgleich globaler Server konfiguriert und befinden sich in der DMZ. Sie bieten somit eine fehlertolerante Lösung für mehrere Sites. Weitere Informationen zu hochverfügbaren Konfigurationen mit mehreren Sites in StoreFront finden Sie unter [Einrichten hochverfügbarer Stores mit mehreren Sites](#).

Zum Verwalten jeder einzelnen Site ist eine eigenständige Studio-Konsole erforderlich. Sites können nicht als Einheit verwaltet werden. Sie können mit Director Benutzer siteübergreifend unterstützen.

Abbildung 2 Eine Konfiguration mit mehreren regionalen Sites und Datacentren.



Datenbankfehlertoleranz

Sep 29, 2015

In diesem Abschnitt wird beschrieben, wie Sie die Fehlertoleranz in einer Bereitstellung erhöhen können, um die Verfügbarkeit essentiell wichtiger Anwendungen und Desktops zu jedem Zeitpunkt sicherzustellen.

Hinweis: Weitere Informationen zum Konfigurieren von Virtual Delivery Agent (VDA) für den Betrieb im Hochverfügbarkeitsmodus finden Sie unter [Sicherstellen des Zugriffs auf Desktops und Anwendungen bei Ausfall der Delivery Controller](#)

Konfigurieren der Datenbankfehlertoleranz

Alle Informationen werden in der Sitekonfigurationsdatenbank gespeichert; Controller kommunizieren nur mit der Datenbank und nicht untereinander. Ein Controller kann entfernt oder ausgeschaltet werden, ohne dass dies Auswirkungen auf die anderen Controller in der Site hat. Dies bedeutet jedoch, dass die Sitekonfigurationsdatenbank eine einzelne Fehlerquelle darstellt. Wenn der Datenbankserver ausfällt, funktionieren die bestehenden Verbindungen zu virtuellen Desktops weiter, bis der Benutzer sich entweder abmeldet oder die Verbindung mit dem virtuellen Desktop trennt. Wenn der Datenbankserver nicht erreichbar ist, können keine neuen Verbindungen hergestellt werden.

Citrix empfiehlt, dass Sie regelmäßig ein Backup der Datenbank durchführen, damit diese in dem Fall, dass der Datenbankserver ausfällt, von dem Backup wiederhergestellt werden kann. Außerdem gibt es einige Hochverfügbarkeitslösungen, die Sie berücksichtigen sollten, um automatisches Failover zu gewährleisten:

- **SQL-Spiegelung:** Dies ist die empfohlene Lösung. Das Spiegeln der Datenbank stellt sicher, dass der automatische Failoverprozess in nur wenigen Sekunden stattfindet, falls der aktive Datenbankserver ausfällt. Benutzer werden in der Regel also nicht beeinträchtigt. Diese Methode ist jedoch teurer als andere Lösungen, da vollständigen SQL-Serverlizenzen auf jedem Datenbankserver erforderlich sind. In gespiegelten Umgebungen kann SQL Server Express nicht verwendet werden.
- **Verwenden der Hochverfügbarkeitsfeatures des Hypervisors:** Bei dieser Methode wird die Datenbank als virtuelle Maschine bereitgestellt und die Hochverfügbarkeitsfeatures des Hypervisors werden verwendet. Diese Lösung ist billiger als das Spiegeln, da sie die bestehende Hostsoftware verwendet und Sie zudem SQL Express verwenden können. Der automatische Failoverprozess ist jedoch langsamer, da eine neue Maschine u. U. eine Weile braucht, bis sie gestartet wird, und dadurch auch die Datenbank. Möglicherweise wird also der Dienst für Benutzer unterbrochen.
- **SQL-Clustering:** Mit dieser Technologie von Microsoft können Sie einem Server automatisch erlauben, die Tasks und Verantwortlichkeiten eines anderen, fehlgeschlagenen Servers zu übernehmen. Es ist jedoch etwas komplizierter, diese Lösung einzurichten. Zudem ist der automatische Failoverprozess in der Regel langsamer als mit anderen Lösungen, wie z. B. SQL-Spiegelung.
- **"AlwaysOn"-Verfügbarkeitsgruppen** sind eine Lösung für Hochverfügbarkeit und Notfallwiederherstellung für Unternehmen, die mit SQL Server 2012 eingeführt wurde. Damit können Sie die Verfügbarkeit für eine oder mehrere Benutzerdatenbanken maximieren. AlwaysOn-Verfügbarkeitsgruppen erfordern, dass die SQL Server-Instanzen auf Windows Server Failover Clustering-Knoten (WSFC) residieren. Weitere Informationen finden Sie unter [AlwaysOn Availability Groups \(SQL Server\)](#).

Hinweis: Die Installation eines Controllers auf einem Knoten in einer SQL-Clustering- oder SQL-Spiegelungsinstallation wird nicht unterstützt.

Konfigurieren einer Site für die Verwendung einer gespiegelten Datenbank

Vor dem Erstellen der Citrix Site müssen zunächst vom Administrator alle Konfigurationsaufgaben mit den SQL Server-Verwaltungstools durchgeführt werden. Beim Ausführen des Assistenten zum Erstellen der Site werden dann die

verbleibenden Aufgaben erledigt.

Bei einer gespiegelten Umgebung werden mindestens zwei SQL-Servermaschinen benötigt (z. B. SQL-Server A und SQL-Server B). SQL Server Express kann nicht für Prinzipal- oder Spiegeldatenbanken verwendet werden.

Verwenden Sie zum Konfigurieren der SQL-Serverdatenbanken die Microsoft SQL Server-Verwaltungstools:

1. Installieren Sie die SQL-Serversoftware auf SQL-Server A und SQL-Server B
2. Erstellen Sie auf SQL-Server A die Datenbank, die als Prinzipaldatenbank verwendet wird (z. B. myDatabaseMirror). Stellen Sie sicher, dass die Datenbank das vollständige Wiederherstellungsmodell und nicht das einfache Modell verwendet. (Das einfache Modell ist standardmäßig konfiguriert, dabei wird jedoch das Transaktionsprotokoll nicht gesichert.)

Stellen Sie sicher, dass die Sortierreihenfolge mit `_CI_AS_KS` endet. (Damit werden Groß- und Kleinschreibung nicht erkannt, Akzente und japanische Schriftzeichen werden erkannt.)

Aktivieren Sie einen Read Committed Snapshot wie unter [How to Enable Read-Committed Snapshot in XenDesktop](#) beschrieben. Es ist wichtig, diese Aktivierung vor dem Spiegeln der Datenbank durchzuführen, um Fehler zu vermeiden.

3. Legen Sie auf SQL-Server A eine Sicherungskopie der Datenbank an und kopieren Sie sie auf SQL-Server B.
4. Stellen Sie auf SQL-Server B die Backupdatei wieder dort (SQL-Server B) her.
5. Beginnen Sie auf SQL-Server A mit der Spiegelung.

Der nächste Schritt hängt davon ab, ob der Citrix Administrator (der Person, die den Assistenten zum Erstellen der Site ausführt) auch über vollständige Datenbankberechtigungen verfügt:

- Wenn der Citrix Administrator Datenbankberechtigungen hat (Datenbank- und Citrix Administrator sind ein- und dieselbe Person), wird alles über Studio abgewickelt:
 1. Der Citrix Administrator erstellt mit Studio eine Site und gibt die Adressen der zuvor erstellten SQL-Serverdatenbank A und deren Namen (myDatabaseMirrorForXD) ein.
 2. Die Datenbankskripts werden automatisch angewendet und die Prinzipal- und die Spiegeldatenbank werden festgelegt.
- Wenn der Citrix Administrator keine Datenbankberechtigungen hat, muss der Citrix Administrator einen Datenbankadministrator hinzuziehen:
 1. Der Citrix Administrator erstellt mit Studio eine Site und gibt die Adressen der zuvor erstellten SQL-Serverdatenbank und deren Namen (myDatabaseMirrorForXD) ein.
 2. Durch Aktivieren von Skript erstellen im Assistenten für die Siteerstellung werden ein Spiegelungs- und ein Primärskript generiert. Der Citrix Administrator gibt diese Skripts an den Datenbankadministrator, der sie dann anwendet (das Spiegelungsskript sollte zuerst angewendet werden). Der Datenbankadministrator muss den Citrix Administrator informieren, wenn die Aufgabe abgeschlossen ist.
 3. Der Citrix Administrator kann jetzt in Studio den Assistenten für die Siteerstellung fortsetzen und abschließen. Die Prinzipal- und die Spiegeldatenbank sind eingerichtet.

Um die Spiegelung nach dem Erstellen der Site zu überprüfen, führen Sie das PowerShell-Cmdlet `get-configdbconnection` aus, damit sichergestellt ist, dass der Failoverpartner in der Verbindungszeichenfolge für die Spiegelung eingerichtet wurde.

Wenn Sie später einen Delivery Controller in einer gespiegelten Datenbankumgebung hinzufügen, verschieben oder entfernen möchten, können Sie entsprechende Anweisungen unter [Hinzufügen, Entfernen oder Verschieben von Controllern](#) anzeigen.

Zugriff auf Desktops und Anwendungen beim Fehlschlagen der Delivery Controllers

Sep 29, 2015

Für den Fall, dass alle Delivery Controller in einer Site ausfallen, können Sie den Virtual Delivery Agent (VDA) für Serverbetriebssystemmaschinen und Desktopbetriebssystemmaschinen für den Hochverfügbarkeitsmodus einrichten, sodass Benutzer weiterhin auf Desktops zugreifen und diese verwenden können. Im Hochverfügbarkeitsmodus akzeptiert der VDA direkte ICA-Verbindungen von Benutzern anstelle von durch den Controller vermittelten Verbindungen.

Hinweis: Dieses Feature ist nur zur Verwendung in dem seltenen Fall, dass die Kommunikation mit allen Controllern fehlschlägt, vorgesehen. Es ist keine Alternative zu anderen Hochverfügbarkeitslösungen, wie z. B. Konfigurieren der Fehlertoleranz bei Datenbanken sowie Sitefailover. Bevor Sie dieses Feature verwenden, beachten Sie die unten stehende Liste der Einschränkungen, da diese sich auf die Sicherheit auswirken.

Wenn die Kommunikation mit dem Controller fehlschlägt, wird der Hochverfügbarkeitsmodus nur initiiert, nachdem ein festgelegter Zeitraum verstrichen ist. Standardmäßig beträgt er 300 Sekunden (5 Minuten). Sie können den Zeitraum jedoch konfigurieren.

Ist der Hochverfügbarkeitsmodus aktiv (aktiviert für 30 Tage), versucht der VDA bis zu 30 Tage lang, die Registrierung bei einem Controller vorzunehmen. In dieser Zeit können Benutzer die Desktops im Hochverfügbarkeitsmodus verwenden. Wenn der Controller zu einem späteren Zeitpunkt wieder verfügbar ist, werden die Desktops registriert und die Benutzersitzungen werden ohne Unterbrechung fortgeführt. Jede weitere Verbindung wird dann jedoch vom Controller wie üblich vermittelt. Wenn die Desktops nach 30 Tagen immer noch nicht bei einem Controller registriert werden konnten, wird das Lauschen auf Verbindungen eingestellt und die Desktops sind nicht mehr verfügbar. Somit hat der Administrator 30 Tage, in denen er die Controllerinfrastruktur reparieren kann. Er sollte also auf den Hochverfügbarkeitsmodus nicht angewiesen sein.

Der Hochverfügbarkeitsmodus eignet sich nur für die Verwendung mit dedizierten Desktops, bei denen die Zuordnung zwischen Benutzer und VDA bekannt ist. Sie können den Hochverfügbarkeitsmodus nicht für die Verwendung mit gepoolten Desktops konfigurieren.

Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab. Gehen Sie folgendermaßen vor, um den Hochverfügbarkeitsmodus zu aktivieren:

1. Legen Sie die Registrierungsschlüssel **HighAvailability** und **HaRegistrarTimeout** fest.
2. Stellen Sie Benutzern eine ICA-Startdatei zur Verfügung, mit der sie direkte ICA-Verbindungen herstellen können. Sie müssen eine ICA-Datei für jeden Benutzer erstellen, der dieses Feature benötigt. Citrix erstellt oder verteilt keine ICA-Dateien für diesen Zweck.

Einrichten der Registrierungsschlüssel

Wenn Sie den VDA so konfigurieren möchten, dass die Ausführung bei Bedarf im Hochverfügbarkeitsmodus erfolgt, fügen Sie folgende Registrierungsschlüssel hinzu. Sie müssen dies nach der Installation des VDAs tun.

1. Fügen Sie den folgenden Registrierungseintrag für HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\VirtualDesktopAgent

hinzu:

Name: HighAvailability

Typ: REG_DWORD

Werte:

1 = Hochverfügbarkeitsmodus aktivieren

0 = Hochverfügbarkeitsmodus deaktivieren

2. Um den Zeitraum zu ändern, in dem der VDA Registrierungsversuche beim Controller unternimmt, bevor der Hochverfügbarkeitsmodus gestartet wird, müssen Sie auch folgenden Registrierungseintrag hinzufügen:

Name: HaRegistrarTimeout

Typ: REG_DWORD

Wert: Anzahl von Sekunden. Der Standardwert ist 300 Sekunden.

3. Starten Sie den virtuellen Desktop neu.

Vorbereiten einer ICA-Startdatei

Damit Benutzer eine direkte ICA-Verbindung mit Desktops herstellen können, stellen Sie ihnen eine ICA-Startdatei zur Verfügung, die sie im Fall, dass der Controller fehlschlägt, verwenden können. Sie müssen eine ICA-Startdatei für jeden Benutzer erstellen, der dieses Feature benötigt. Citrix erstellt und verteilt keine ICA-Dateien für diesen Zweck. Weitere Informationen zum Erstellen von ICA-Dateien finden Sie unter <http://support.citrix.com/article/CTX127392>.

Teilen Sie den Benutzern mit, in welchen Fällen die Verwendung der ICA-Startdatei angebracht ist und wie sie auf die Datei zugreifen können.

Beschränkungen des Hochverfügbarkeitsmodus

Der Hochverfügbarkeitsmodus eignet sich nur für die Verwendung mit dedizierten Desktops. Sie können ihn nicht für die Verwendung mit gepoolten Desktops konfigurieren.

Im Hochverfügbarkeitsmodus stehen einige Features nicht zur Verfügung, Dazu gehören:

- Benutzerroaming: Wenn ein Benutzergerät bereits mit dem Desktop verbunden ist, können Benutzer keine Verbindung über ein anderes Gerät herstellen.
- Energieverwaltung: Wenn der Desktop hochfährt, versucht er sich zu registrieren, schlägt fehl und tritt nach dem Timeout in den Hochverfügbarkeitsmodus ein.
- Vom Delivery Controller stammende Richtlinien: Richtlinien, die auf dem Controller erstellt wurden, wie z. B. diejenigen zum Steuern der Clientlaufwerkzuordnung und des Zugriffs auf die Zwischenablage, funktionieren nicht, da keine Verbindung mit dem Controller besteht. Richtlinien, die vom Domänencontroller stammen, und lokale Gruppenrichtlinien sind nicht betroffen. Beachten Sie, dass von einer früheren Registrierung stammende Richtlinien erhalten bleiben und angewendet werden, so dass möglicherweise veraltete Richtlinien wirksam werden.
- NetScaler Gateway und Remotezugriff:

Der Hochverfügbarkeitsmodus dauert nur bis zu 30 Tage. Danach sind die Desktops nicht mehr verfügbar.

Security

Sep 29, 2015

In diesem Abschnitt wird Folgendes beschrieben:

- Allgemeine bewährte Methoden zur Gewährleistung der Sicherheit beim Verwenden dieses Releases sowie sicherheitsrelevante Unterschiede zwischen diesem Release und einer konventionellen Computerumgebung
- Verwalten von Benutzerprivilegien
- Bereitstellungsszenarios und ihre Auswirkungen auf die Sicherheit
- Remote-PC-Zugriff

Möglicherweise muss Ihre Organisation bestimmte Sicherheitsstandards einhalten, um den gesetzlichen Anforderungen zu genügen. In diesem Dokument wird dieses Thema nicht behandelt, da sich Sicherheitsstandards mit der Zeit ändern. Aktuelle Informationen über Sicherheitsstandards und Citrix Produkte finden Sie unter <http://www.citrix.com/security/>.

Bewährte Methoden zur Gewährleistung der Sicherheit

Halten Sie stets alle Computer in der Umgebung mit Sicherheitspatches auf dem neuesten Stand. Ein Vorteil besteht darin, dass Thin Clients als Terminals verwendet werden können. Das erleichtert diese Aufgabe.

Schützen Sie alle Maschinen in der Umgebung mit Antivirensoftware.

Schützen Sie alle Maschinen in der Umgebung mit Umkreisfirewalls, u. a. bei Bedarf auch an Grenzen von Enklaven.

Wenn Sie eine konventionelle Umgebung zu diesem Release migrieren, müssen Sie ggf. eine vorhandene Umkreisfirewall neu positionieren oder neue Umkreisfirewalls hinzufügen. Beispiel: Zwischen einem konventionellen Client und einem Datenbankserver im Datenzentrum befindet sich eine Umkreisfirewall. Bei diesem Release muss diese Umkreisfirewall aber so platziert werden, dass sich der virtuelle Desktop und das Benutzergerät auf der einen Seite der Firewall befinden und die Datenbankserver und Controller im Rechenzentrum auf der anderen Seite. Sie sollten deshalb erwägen, im Rechenzentrum eine Enklave für die in diesem Release verwendeten Server und Controller zu erstellen. Sie sollten zudem in Betracht ziehen, einen Schutz zwischen dem Benutzergerät und dem virtuellen Desktop zu installieren.

Alle Maschinen in der Umgebung müssen durch eine persönliche Firewall geschützt werden. Wenn Sie einen Virtual Delivery Agent (VDA) installieren, können Sie die erforderlichen Ports für Komponenten und Features so einrichten, dass sie automatisch geöffnet werden, sobald der Windows-Firewalldienst erkannt wird (auch wenn die Firewall nicht aktiviert ist). Sie können die Firewallports auch manuell konfigurieren. Wenn Sie eine andere Firewall verwenden, muss diese manuell konfiguriert werden.

Hinweis: Da die TCP-Ports 1494 und 2598 für ICA und CGP verwendet werden, sind sie normalerweise an der Firewall geöffnet, damit Benutzer außerhalb des Rechenzentrums auf sie zugreifen können. Citrix empfiehlt, dass diese Ports nicht für etwas Anderes verwendet werden, damit administrative Benutzeroberflächen nicht versehentlich gefährdet werden. Die Ports 1494 und 2598 sind bei der Internet Assigned Number Authority (siehe <http://www.iana.org/>) offiziell registriert. Die gesamte Netzwerkkommunikation sollte Ihren Sicherheitsrichtlinien gemäß angemessen gesichert und verschlüsselt werden. Sie können die gesamte Kommunikation zwischen Microsoft Windows-Computern mit IPSec sichern. Weitere Informationen hierzu finden Sie in der Dokumentation zum Betriebssystem. Die Kommunikation zwischen Benutzergeräten und Desktops ist außerdem mit Citrix SecureICA gesichert, das in der Standardeinstellung 128-Bit-Verschlüsselung verwendet. Sie können beim Erstellen oder Aktualisieren einer Zuweisung SecureICA konfigurieren (siehe [Sichern von Bereitstellungsgruppen](#)).

Verwalten von Benutzerprivilegien

Sie sollten den Benutzern nur die Privilegien einräumen, die sie benötigen. Microsoft Windows-Privilegien können weiterhin in der üblichen Weise auf Desktops angewendet werden: Konfigurieren Sie Privilegien mit "Zuweisung von Benutzerrechten" und Gruppenmitgliedschaften mit einer Gruppenrichtlinie. Der Vorteil dieses Release besteht darin, dass einem Benutzer Administratorrechte für einen Desktop eingeräumt werden können, ohne ihm auch die physische Kontrolle über den Computer, auf dem der Desktop gespeichert ist, zu gewähren.

Beachten Sie beim Planen von Desktopprivilegien Folgendes:

- Standardmäßig wird nicht berechtigten Benutzern beim Herstellen einer Verbindung mit einem Desktop die Zeitzone des Systems, auf dem der Desktop ausgeführt wird, statt der Zeitzone ihres eigenen Benutzergerätes angezeigt. Weitere Informationen dazu, wie Sie Benutzern erlauben, ihre Ortszeit beim Verwenden von Desktops anzuzeigen, finden Sie unter [Konfigurieren der Zeitzoneneinstellungen](#).
- Ein Benutzer mit Administratorrechten auf einem Desktop hat Vollzugriff auf diesen Desktop. Wenn ein Desktop ein gepoolter Desktop und kein dedizierter Desktop ist, muss dem Benutzer von allen anderen Benutzern dieses Desktops, einschließlich zukünftiger Benutzer, vertraut werden. Alle Benutzer des Desktops müssen sich des potenziellen permanenten Risikos für ihre Datensicherheit bewusst sein, die diese Situation mit sich bringt. Diese Überlegung trifft nicht auf dedizierte Desktops zu, die nur einen einzelnen Benutzer haben. Dieser Benutzer sollte kein Administrator auf einem anderen Desktop sein.
- Ein Benutzer mit Administratorrechten auf einem Desktop kann auf diesem Desktop generell Software installieren, einschließlich potenziell schädlicher Software. Zudem kann der Benutzer u. U. den Datenverkehr in allen mit dem Desktop verbundenen Netzwerken überwachen und steuern.

Auswirkungen von Bereitstellungsszenarios auf die Sicherheit

Ihre Benutzerumgebung kann aus Benutzergeräten bestehen, die von Ihrer Organisation nicht verwaltet werden und dem Vollzugriff der jeweiligen Benutzer unterliegen. Sie kann aber auch aus Benutzergeräten bestehen, die von Ihrer Organisation verwaltet werden. Die Sicherheitsüberlegungen für diese beiden Umgebungen sind generell unterschiedlich.

Verwaltete Benutzergeräte

Verwaltete Benutzergeräte unterliegen einer administrativen Steuerung. Sie werden entweder von Ihnen gesteuert oder von einer anderen Organisation, der Sie vertrauen. Sie können Benutzergeräte konfigurieren und Benutzern direkt bereitstellen. Alternativ können Sie Terminals bereitstellen, auf denen ein einzelner Desktop im Vollbildmodus ausgeführt wird. Sie sollten die oben beschriebenen allgemeinen bewährten Sicherheitsmethoden für alle verwalteten Benutzergeräte anwenden. Dieses Release bietet den Vorteil, dass nur ganz wenig Software auf einem Benutzergerät erforderlich ist.

Ein verwaltetes Benutzergerät kann für die Verwendung im Vollbildmodus oder im Fenstermodus eingerichtet werden.

- Wenn ein Benutzergerät für die Verwendung im Vollbildmodus konfiguriert ist, können Benutzer sich mit dem üblichen Bildschirm "Anmelden an Windows" anmelden. Dieselben Anmeldeinformationen des Benutzers werden dann zum automatischen Anmelden für dieses Release verwendet.
- Wenn Benutzergeräte so konfiguriert sind, dass Benutzern ihre Desktops in einem Fenster angezeigt werden, melden sich die Benutzer zunächst beim Benutzergerät an. Anschließend melden sie sich über die in diesem Release bereitgestellte Website bei diesem Release an.

Nicht verwaltete Benutzergeräte

Wenn Benutzergeräte nicht von einer vertrauenswürdigen Organisation verwaltet werden, kann nicht von einer

administrativen Steuerung ausgegangen werden. Beispiel: Sie erlauben Benutzern, sich ihre eigenen Geräte zu besorgen und sie zu konfigurieren, doch die Benutzer halten sich u. U. nicht an die oben beschriebenen generellen optimalen Sicherheitsverfahren. Dieses Release hat den Vorteil, nicht verwalteten Benutzergeräten Desktops sicher bereitstellen zu können. Diese Geräte sollten jedoch einen grundlegenden Antivirenschutz haben, um Keylogger und ähnliche Angriffe auf Benutzereingaben abzuwehren.

Überlegungen zum Datenspeicher

Mit diesem Release können Sie verhindern, dass Benutzer Daten auf Benutzergeräten speichern, die sie selbst physisch steuern können. Sie müssen dennoch bedenken, welche Auswirkungen es haben kann, wenn Benutzer Daten auf Desktops speichern. Im Allgemeinen sollten Benutzer keine Daten auf Desktops speichern. Daten sollten an einem Ort gespeichert werden, an dem sie entsprechend geschützt werden können, wie z. B. auf Dateiservern, Datenbankservern oder in anderen Repositories.

Möglicherweise enthält Ihre Desktopumgebung verschiedene Desktoptypen, wie gepoolte und dedizierte Desktops.

- Benutzer sollten zu keiner Zeit Daten auf Desktops speichern, die für andere Benutzer freigegeben sind, wie z. B. gepoolte Desktops.
- Wenn Benutzer Daten auf dedizierten Desktops speichern, sollten diese Daten entfernt werden, wenn der Desktop zu einem späteren Zeitpunkt anderen Benutzern zugänglich gemacht wird.

Remote-PC-Zugriff

Mit Remote-PC-Zugriff werden die folgenden Sicherheitsfeatures implementiert:

- Die Verwendung von Smartcards wird unterstützt.
- Bei Verbindung einer Remotesitzung wird der Monitor des Büro-PCs leer angezeigt.
- Remote-PC-Zugriff leitet alle Tastatur- und Mauseingaben in die Remotesitzung um, ausgenommen Strg + Alt + Entf, USB-aktivierte Smartcards und biometrische Geräte.
- SmoothRoaming wird nur für einen einzelnen Benutzer unterstützt.
- Wenn ein Benutzer über eine Remotesitzung mit einem Büro-PC verbunden ist, kann nur dieser Benutzer den lokalen Zugriff auf den Büro-PC wiederaufnehmen. Zum Wiederaufnehmen des lokalen Zugriffs muss der Benutzer Strg + Alt + Entf auf dem lokalen PC drücken und sich dann mit denselben Anmeldeinformationen wie für die Remotesitzung anmelden. Er kann zudem auch über eine Smartcard oder biometrische Geräte wieder lokal zugreifen, wenn das System die entsprechende Anmeldeinformationsanbieter-Integration besitzt.
Hinweis: Das Standardverhalten kann über die schnelle Benutzerumschaltung über Gruppenrichtlinienobjekte oder durch Bearbeiten der Registrierung außer Kraft gesetzt werden.
- Standardmäßig unterstützt Remote-PC-Zugriff die automatische Zuweisung von mehreren Benutzern zu einem VDA. Unter XenDesktop 5.6 Feature Pack 1 konnten Administratoren dieses Verhalten mit dem PowerShell-Skript RemotePCAccess.ps1 außer Kraft setzen. Dieses Release verwendet einen Registrierungseintrag, mit dem mehrere automatische Remote-PC-Zuweisungen zugelassen oder abgelehnt werden; diese Einstellung gilt für die gesamte Site. Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab.

So beschränken Sie die automatische Zuweisung auf einen einzelnen Benutzer:

1. Legen Sie den folgenden Registrierungseintrag auf jedem Controller in der Site fest:
HKEY_LOCAL_MACHINE\Software\Citrix\DesktopServer

Name: AllowMultipleRemotePCAssignments

Typ: REG_DWORD

Daten: siehe Diagramm

Wert	Ergebnis
0	Zuweisung mehrerer Benutzer deaktivieren.
1	Zuweisung mehrerer Benutzer aktivieren. Dies ist der Standardwert.

2. Liegen bereits Benutzerzuweisungen vor, entfernen Sie diese mit SDK-Befehlen, damit der VDA anschließend für eine einzelne automatische Zuweisung zur Verfügung steht. Weitere Informationen zum Verwenden des SDKs finden Sie unter [Informationen zum XenApp- und XenDesktop-SDK](#).
 1. Entfernen Sie alle zugewiesenen Benutzer aus dem VDA:\$machine.AssociatedUserNames | %{ Remove-BrokerUser-Name \$_ -Machine \$machine
 2. Entfernen Sie den VDA aus der Desktopgruppe:\$machine | Remove-BrokerMachine -DesktopGroup \$desktopGroup
3. Starten Sie den physischen Büro-PC neu.

Hosten von Desktops und Anwendungen

Sep 29, 2015

Um gehostete Anwendungen und Desktops für die Benutzer über Studio bereitzustellen, sollten Sie die folgenden Umgebungseigenschaften evaluieren.

- Unterstützte Benutzertypen
 - Benutzer führen vordefinierte Aufgaben aus, es wird keine Personalisierung oder Offline-Zugriff benötigt (z. B. Callcenter).
 - Benutzer führen verschiedene Aufgaben aus und dürfen ggf. ihre eigenen Anwendungen installieren und verwalten (z. B. interne und externe Auftragnehmer, Drittanbieter oder andere vorläufige Teammitglieder).
 - Benutzer arbeiten von zu Hause oder von anderen Orten und benötigen Zugriff auf bestimmte Software oder Daten auf ihren Desktops, um ihre Aufgaben remote durchzuführen.
 - Benutzer benötigen Zugriff auf lokal installierte Anwendungen direkt auf ihren physischen Geräten und auch auf ihren virtuellen Desktops.
 - Benutzer müssen Windows-Anwendungen ohne Netzwerkzugriff auf ihrem Windows-Gerät starten.
- Benötigte Bereitstellungsumgebung
 - Kostengünstige, serverbasierte Bereitstellung, um die Kosten für die Bereitstellung von Anwendungen für eine große Anzahl von Benutzern gering zu halten, und gleichzeitig eine sichere High-Definition-Benutzererfahrung zu bieten.
 - Zentral verwaltete Anwendungen und Desktops, die vom Benutzer nicht geändert werden können.
 - Adäquate Ressourcen für Benutzer, die bei der Anmeldung eine einzelne Maschine verbrauchen und Anwendungen und Daten bearbeiten und speichern.
 - Zugriff auf zentral verwaltete Windows App-V-Anwendungen und deren Bereitstellung auf Benutzergeräten.
- Typ der bereitzustellenden Anwendungen
 - Jede Windows-Anwendung.
 - Anwendungen, die möglicherweise nicht gut mit anderen Anwendungen funktionieren oder mit dem Betriebssystem interagieren, z. B. .NET Framework.
 - Anwendungen, die auf älteren Betriebssystemen (z. B. Windows XP oder Windows Vista) und älteren Architekturen (z. B. 32 Bit oder 16 Bit) ausgeführt werden.
 - Anwendungen, die den Remotebenutzern über einen Bürocomputer bereitgestellt werden.
 - Spezial- oder Nischenanwendungen, die noch nicht virtualisiert sind, z. B. AutoCAD, oder Anwendungen die auf bestimmte lokale Hardware, z. B. DVD-Brenner, zugreifen müssen.
 - Von Microsoft unterstützte App-V-Anwendungen
- Speicherort der von den Benutzern benötigten Anwendungen
 - Physische oder virtuelle Maschinen in einem Datenzentrum
 - Ältere Betriebssysteme, z. B. Windows XP oder Windows Vista
 - Windows App-V-Server
 - Remotebenutzermaschinen.

Wenn Sie die Benutzer- und Bereitstellungsanforderungen festgelegt haben, finden Sie weitere Informationen zur Auswahl der Bereitstellungsmethode von Anwendungen und Desktops auf Benutzergeräten über Studio unter [Wählen einer Anwendungs- und Desktop-Bereitstellungsmethode](#).

Wählen einer Anwendungs- und Desktop-Bereitstellungsmethode

Die Methode, mit der Sie den Benutzern Anwendungen oder Desktops bereitstellen, hängt von den Typen der Anwendungen und Desktops ab, die Sie hosten, den verfügbaren Systemressourcen, dem Typen der Benutzer und der

angestrebten Benutzererfahrung.

Wählen Sie zwischen den folgenden Anwendungs- und Desktop-Bereitstellungsmethoden:

- Gehostete Anwendungen oder Desktops auf:
 - Serverbetriebssystemmaschinen
 - Desktopbetriebssystemmaschinen
- Gehostete Desktops bei Remote-PC-Zugriff
- Gehostete Anwendungen über:
 - Lokaler App-Zugriff: Integration von lokalen installierten Anwendungen des Benutzers sowie von gehosteten Anwendungen in einer gehosteten Desktopumgebung.
 - Microsoft Application Virtualization-Anwendungen (App-V): Stellt App-V-Anwendungen von einem virtuellen Anwendungsserver als gehostete Anwendungen auf Benutzergeräten bereit. Wenn die Anwendungen wie unter [Microsoft Application Virtualization](#) beschrieben konfiguriert wurden, können sie Anwendungsbereitstellungsgruppen oder Desktop- und Anwendungsbereitstellungsgruppen hinzugefügt werden. Beachten Sie Folgendes:
 - Versionen vor App-V 5.0 werden nicht unterstützt.
 - Der App-V-Client 5,0 bietet keinen Offlinezugriff auf Anwendungen.
 - Verwenden Sie die folgenden Versionskombinationen für App-V- und XenApp-/XenDesktop-Komponenten:

App-V version	XenDesktop-/XenApp-Version	
	Delivery Controller	Virtual Delivery Agent (VDA)
5.0 R5TM	XenDesktop 7.0, 7.1, 7.5 XenApp 7.5	7.0, 7.1, 7.5
5.0 Service Pack 1	XenDesktop 7.0, 7.1, 7.5 XenApp 7.5	7.0, 7.1, 7.5
5.0 Service Pack 2	XenDesktop 7.0, 7.1, 7.5 XenApp 7.5	7.1, 7.5

Unabhängig von der gewählten Methode werden die Anwendungen und Desktops den Benutzergeräten virtuell von einer Maschine in einer Bereitstellungsgruppe bereitgestellt, der der Benutzer zugewiesen wurde.

Bereitstellungsoptionen

Nachfolgend wird beschrieben, welche Situationen, Benutzer und Überlegungen für die Auswahl der Anwendungs- und Desktop-Bereitstellungsmethode in Studio zu berücksichtigen sind.

Serverbetriebssystemmaschinen

Anwendungsfall	Ziel
----------------	------

	Kostengünstige, serverbasierte Bereitstellung, um die Kosten für die Bereitstellung von Anwendungen für eine große Anzahl von Benutzern gering zu halten, und gleichzeitig eine sichere High-Definition-Benutzererfahrung zu bieten. Benutzer Benutzer führen vordefinierte Aufgaben aus, es wird keine Personalisierung oder kein Offlinezugriff auf Anwendungen benötigt. Hierzu können aufgabenorientierte Mitarbeiter, wie z. B. Callcenter- und Einzelhandelsarbeitskräfte gehören, oder Benutzer, die Arbeitsstationen gemeinsam verwenden. Anwendungstypen Beliebige Anwendung.
Vorteile und Überlegungen	Vorteile Verwaltbare und skalierbare Lösung für das Datenzentrum. Kosteneffektivste Lösung für die Anwendungsbereitstellung. Gehostete Anwendungen werden zentral verwaltet und Benutzer können die Anwendung nicht ändern, wodurch eine konsistente, sichere und zuverlässige Benutzererfahrung bereitgestellt wird. Überlegungen Benutzer müssen online sein, um auf ihre Anwendungen zuzugreifen.
Benutzererfahrung	Benutzer fordern eine oder mehrere Anwendungen von StoreFront über ihr Startmenü oder eine von Ihnen vorgegebene URL an. Anwendungen werden virtuell bereitgestellt und in High Definition auf Benutzergeräten angezeigt. Abhängig von den Profileinstellungen werden Benutzeränderungen gespeichert, wenn die Anwendungssitzung des Benutzers beendet wird. Andernfalls werden die Änderungen werden gelöscht.
Verarbeiten, Hosten und Bereitstellen von Anwendungen	Prozess Die Anwendungsverarbeitung findet auf den Hostingmaschinen statt, nicht auf den Benutzergeräten. Die Hostingmaschine kann eine physische oder eine virtuelle Maschine sein. Host Anwendungen und Desktops sind auf einer Serverbetriebssystemmaschine gespeichert. Maschinen werden über Maschinenkataloge verfügbar gemacht.

	Bereitstellungsgruppe Maschinen in Maschinenkatalogen sind in Bereitstellungsgruppen organisiert, die Benutzergruppen dieselben Anwendungen bereitstellen. Serverbetriebssystemmaschinen unterstützen Folgendes: • Desktop- und Anwendungsbereitstellungsgruppen, die sowohl Desktops als auch Anwendungen hosten. • Anwendungsbereitstellungsgruppen, die nur Anwendungen hosten
Sitzungsverwaltung und -zuweisung	Sitzungen Auf Serverbetriebssystemmaschinen werden mehrere Sitzungen auf einer einzelnen Maschinen ausgeführt, über die mehrere Anwendungen und Desktops an mehrere, gleichzeitig verbundene Benutzer bereitgestellt werden. Jeder Benutzer benötigt eine einzelne Sitzung, um die gehosteten Anwendungen auszuführen. Beispiel: Ein Benutzer meldet sich an und fordert eine Anwendung an. Eine der Sitzungen auf dieser Maschine ist für die anderen Benutzer nicht mehr verfügbar. Ein zweiter Benutzer meldet sich an und fordert eine Anwendung an, die von dieser Maschine gehostet wird. Eine zweite Sitzung auf derselben Maschine ist damit jetzt nicht verfügbar. Wenn beide Benutzer weitere Anwendungen anfordern, werden keine zusätzlichen Sitzungen benötigt, da ein Benutzer mehrere Anwendungen in der gleichen Sitzung ausführen kann. Wenn zwei weitere Benutzer sich anmelden und Desktops anfordern, und zwei Sitzungen auf derselben Maschine verfügbar sind, hostet diese eine Maschine nun vier Sitzungen für vier verschiedene Benutzer. Zufällige Maschinenzuweisung In der Bereitstellungsgruppe, der ein Benutzer zugewiesen ist, wird eine Maschine auf einem Server mit der geringsten Last ausgewählt. Ein Computer mit Sitzungsverfügbarkeit wird nach dem Zufallsprinzip zugewiesen und stellt einem Benutzer bei der Anmeldung Anwendungen bereit.

Desktopbetriebssystemmaschinen

Anwendungsfall	Ziel Eine clientbasierte Anwendungsbereitstellungslösung, die eine sichere, zentrale Verwaltung bietet und eine große Anzahl von Benutzern pro Hostserver (oder Hypervisor) unterstützt und Benutzern Anwendungen bereitstellt, die problemlos in High Definition angezeigt werden. Benutzer Interne und externe Auftragnehmer, Drittanbietern und andere vorläufige Teammitglieder. Die Benutzer benötigen keinen Offlinezugriff auf gehostete Anwendungen. Anwendungstypen
-----------------------	---

	Anwendungen, die möglicherweise nicht gut mit anderen Anwendungen funktionieren oder mit dem Betriebssystem interagieren, z. B. .NET Framework. Dieser Typ von Anwendungen eignet sich gut für das Hosting auf virtuellen Maschinen. Anwendungen, die auf älteren Betriebssystemen (z. B. Windows XP oder Windows Vista) und älteren Architekturen (z. B. 32 Bit oder 16 Bit) ausgeführt werden. Durch Isolieren jeder Anwendung auf einer eigenen virtuellen Maschine werden die anderen Benutzer bei Ausfall einer Maschine nicht beeinträchtigt.
Vorteile und Überlegungen	Vorteile Anwendungen und Desktops auf dem Masterimage werden sicher verwaltet, gehostet und auf Maschinen im Datenzentrum ausgeführt, womit eine kosteneffektivere Lösung für die Anwendungsbereitstellung bereitgestellt wird. • Die Benutzer können bei der Anmeldung willkürlich einer Maschine in einer Bereitstellungsgruppe zugewiesen werden, die für das Hosting einer Anwendung konfiguriert ist. • Sie können auch einem einzelnen Benutzer eine einzelne Maschine für die Anwendungsbereitstellung jedes Mal statisch zuweisen, wenn sich der Benutzer anmeldet. Bei statisch zugewiesenen Maschinen kann der Benutzer eigene Anwendungen auf der virtuellen Maschine installieren und verwalten. Überlegungen Das Ausführen mehrerer Sitzungen auf Desktopbetriebssystemmaschinen wird nicht unterstützt. Daher beansprucht jeder Benutzer bei der Anmeldung eine einzelne Maschine innerhalb einer Bereitstellungsgruppe und der Zugriff auf die Anwendungen muss online erfolgen. Bei dieser Methode werden die Serverressourcen für die Verarbeitung von Anwendungen sowie der Speicher für die persönlichen vDisks der Benutzer erhöht.
Benutzererfahrung	Die gleiche nahtlose Anwendungserfahrung wie mit gehosteten, freigegebenen Anwendungen auf Serverbetriebssystemmaschinen.
Verarbeiten, Hosten und Bereitstellen von Anwendungen	Prozess Wie bei Serverbetriebssystemmaschinen, nur dass es sich um virtuelle Desktopbetriebssystemmaschinen handelt. Host Wie bei Serverbetriebssystemmaschinen, nur dass es sich um virtuelle Desktopbetriebssystemmaschinen handelt. Bereitstellungsgruppe Wie bei Serverbetriebssystemmaschinen, allerdings können Desktopbetriebssystemmaschinen nur in einer Desktop-Bereitstellungsgruppe bestehen.

Sitzungsverwaltung und -zuweisung	Sitzungen Desktopbetriebssystemmaschinen führen eine Desktopsitzung von einer Maschine aus. Nur beim Zugriff auf Anwendungen: Ein Benutzer kann mehrere Anwendungen verwenden (und ist nicht auf eine Anwendung eingeschränkt), da das Betriebssystem jede Anwendung als eine neue Sitzung ansieht. Zufällige und statische Maschinenzuweisungen Nach der Anmeldung können Benutzer in der Bereitstellungsgruppe, der sie zugewiesen wurden, auf Folgendes zugreifen: • Statisch zugewiesene Maschine, sodass sich der Benutzer jedes Mal an derselben Maschine anmeldet. • Zufällig zugewiesene Maschine, die je nach Sitzungsverfügbarkeit ausgewählt wird.
--	---

Remote-PC-Zugriff

Anwendungsfall	• Ziel: Mitarbeiter mit sicheren Remotezugriff auf einem physischen Computer ohne VPN. Beispiel: Der Benutzer greift auf den physischen Desktop-PC von zu Hause oder über einen öffentlichen WiFi-Hotspot zu. Abhängig vom Standort möchten Sie u. U. das Drucken oder das Kopieren und Einfügen außerhalb des Desktops einschränken. Unterstützung von "Bring your own device" ohne die Desktopimages in das Datacenter zu migrieren. Benutzer Mitarbeiter oder Auftragnehmer, die ggf. von zu Hause arbeiten können, aber Zugriff auf bestimmte Software oder Daten ihrer Unternehmensdesktops benötigen, um ihre Arbeit remote zu erledigen. Host Wie bei Desktopbetriebssystemmaschinen. Anwendungstypen Anwendungen, die von einem Bürocomputer bereitgestellt werden und problemlos in High Definition auf dem Gerät des Remotebenutzers angezeigt werden.
Vorteile und Überlegungen	• Verwaltbare und skalierbare Lösung für das Datacenter. Profitiert von vorhandenen physischen Desktopinvestitionen. • Sie können optional zulassen, dass Benutzer ihre Remotemaschinen über das Wake-On-LAN-Feature von Microsoft Configuration Manager starten. Andernfalls müssen Remotemaschinen immer online sein, damit Benutzer auf ihre Anwendungen zugreifen

	können.
Benutzererfahrung	Benutzer fordern eine Verbindung zu ihren Bürocomputern an.
Verarbeiten, Hosten und Bereitstellen von Anwendungen	Prozess Die Anwendungsverarbeitung findet auf den Bürocomputern statt, nicht auf den Geräten der Remotebenutzer. Host Wie bei Desktopbetriebssystemmaschinen. Bereitstellen Wie bei Desktopbetriebssystemmaschinen.
Benutzerverwaltung und -zuweisung	Remote-PC-Zugriff-Maschinen nutzen eine einzelne Verbindung einer einzelnen Maschine, die Zugriff auf einen Bürocomputer des Benutzers gibt.

Lokaler App-Zugriff

Anwendungsfall	Ziel Integration von lokal installierten Anwendungen der Benutzer und von gehosteten Anwendungen in einer gehosteten Desktopumgebung. Benutzer Direkter Zugriff vom virtuellen Desktop auf Anwendungen, die lokal auf dem physischen Laptop, PC oder einem anderen Gerät installiert sind. Anwendungstypen • Videokonferenzsoftware, z. B. GoToMeeting. • Spezial- oder Nischenanwendungen, die noch nicht virtualisiert sind, z. B. AutoCAD. • Anwendungen, die auf lokale Hardware zugreifen müssen, z. B. DVD-Brenner.
Vorteile	Stellt eine flexible Anwendungsbereitstellungslösung dar. Wenn Benutzer lokale Anwendungen haben, die Sie nicht virtualisieren können oder die IT nicht verwaltet, verhalten sich diese Anwendungen weiterhin so, als ob sie auf einem virtuellen Desktop installiert wären. Da die Anwendungen auf dem Benutzergerät installiert sind und dort ausgeführt werden, werden auf dem Netzwerk weniger Speicherplatz und Verarbeitungsressourcen benötigt. Die Benutzer können auf alle ihre Anwendungen von einem Ort aus zugreifen. Das bedeutet, dass sie nicht zwischen einem virtuellen oder physischen Desktops hin- und her wechseln müssen, um eine Anwendung zu verwenden.

Benutzererfahrung	Benutzer können die Anwendung starten und unmittelbar anzeigen: • Über eine Verknüpfung auf ihrem virtuellen Desktop • Direkt auf ihrem physischen Desktop Anwendungsänderungen werden auf den Benutzergeräten gespeichert.
Verarbeiten, Hosten und Bereitstellen von Anwendungen	Verarbeiten und Hosten Anwendungen werden installiert und lokal auf dem Benutzergerät ausgeführt. Bereitstellungsgruppe Der Benutzer startet die lokal installierte Anwendung mit einer Verknüpfung zu lokal installierten Anwendungen auf den virtuellen Desktops des Benutzers.
Sitzungsverwaltung und -zuweisung	Sitzungen Wenn ein Benutzer die Verbindung mit einem virtuellen Desktop trennt, bleiben die lokal ausgeführten Anwendungen auf der lokalen Benutzermaschine in lokalen Anwendungsfenstern. Vom Benutzer gehostete Anwendungen bleiben ebenso auf der lokalen Benutzermaschine. Wenn sich ein Benutzer von der Sitzung abmeldet oder den virtuellen Desktop herunterfährt, werden alle lokal ausgeführten Anwendungsfenster in der Sitzung geschlossen, genauso wie bei jeder anderen lokalen Anwendung.

App-V-Anwendungen

Anwendungsfall	Ziel Windows App-V-Anwendungen zentral verwalten und sie auf einem Gerät des Benutzers mit Citrix Receiver bereitstellen. Benutzer Benutzer müssen Windows-Anwendungen ohne Netzwerkzugriff auf ihrem Windows-Gerät starten. Anwendungstypen Alle von Microsoft unterstützten App-V-Anwendungen
Vorteile und Überlegungen	Vorteile Wird sowohl auf Serverbetriebssystem- als auch Desktopbetriebssystemmaschinen unterstützt.

	Dies ist eine skalierbare und kostengünstige Lösung, bei der Anwendungen von mehreren App-V-Benutzern von einer einzelnen Maschine gemeinsam verwendet werden. Überlegungen Versionen vor App-V 5.0 werden nicht unterstützt. Der Offlinezugriff auf Anwendungen wird von App-V-Client 5 nicht unterstützt.
Benutzererfahrung	Wie bei gehosteten, freigegebenen Anwendungen, mit dem Unterschied, dass der Benutzer die Anwendung immer mit Citrix Receiver anfordert, und sie dann unmittelbar an das Gerät des Benutzers übertragen wird.
Verarbeiten, Hosten und Bereitstellen von Anwendungen	Verarbeiten und Hosten Anwendungen werden von App-V-Servern gestreamt und auf Maschinen im Datacenter ausgeführt. Bereitstellungsgruppe Wenn ein Benutzer eine Anwendung mit Receiver anfordert, wird diese Anwendung den Benutzern von Desktop- und Anwendungsbereitstellungsgruppen bereitgestellt, denen Benutzern zugewiesen sind.
Sitzungsverwaltung und -zuweisung	Sitzungen Auf jeder Benutzermaschine wird eine einzige Sitzung ausgeführt, in der der Benutzer alle zugewiesenen Anwendungen ausführen kann.

Auswerten von Anwendungen

Zum Evaluieren und zum Beheben von Kompatibilitätsproblemen für die Bereitstellung von Desktop- und Webanwendungen oder für Upgrades des Betriebssystems stellt die AppDNA-Anwendungsmigrationssoftware eindeutige Informationen bereit, um Ihre Bereitstellung zu beschleunigen.

Mit AppDNA ermitteln Sie die Auswirkungen von Anwendungsproblemen und vorgeschlagenen Implementierungen auf die Umgebung, Geräte und Benutzer.

Das AppDNA mit Microsoft System Center Configuration Manager und Active Directory integriert ist, können Sie Anwendungen, die sich im Konfigurations-Manager befinden, schnell importieren und die Optionen ohne Benutzereingriff beim automatisierten Neuverpacken verwenden. Sie haben außerdem folgende Möglichkeiten:

- Erstellen von Anwendungsgruppen, die Ihre Organisationsstruktur in AppDNA darstellen
- Anzeigen von Berichten und Beurteilungsberichten gemäß dieser Gruppen

AppDNA führt eine statische Analyse der Dateien jeder Anwendung, der Registrierungseinträge und der API-Verwendung (d. h. die Anwendungs-DNA) durch, um zu beurteilen, ob die Anwendung für Folgendes geeignet ist:

- Zahlreiche Plattformen, u. a. neue Versionen von Windows Server und Clients
- 64-Bit-Serverumgebung

- Freigegebene serverbasierte Bereitstellung
- Mehrere gleichzeitige Benutzer in separaten Sitzungen
- Anwendungsvirtualisierung mit App-V

Wenn AppDNA mögliche Anwendungsprobleme bei einer bestimmten Plattform erkennt, werden detaillierte Informationen über die zur Problembehebung nötigen Schritte angezeigt.

Die gesamte Dokumentation zu AppDNA, einschließlich Systemanforderungen, Installationsanweisungen und erste Schritte, finden Sie unter [AppDNA](#) in den eDocs.

Verbessern der Benutzererfahrung für Mobilgeräte

XenApp und XenDesktop bieten eine hervorragende Benutzererfahrung für mobile Benutzer, oftmals ohne dass eine spezielle Konfiguration erforderlich ist. Integrieren Sie XenApp und XenDesktop in anderen Citrix Produkten, um den mobilen Mitarbeitern zusätzliche Features zu bieten.

Feature:	Bereitgestellt von:
High Definition-Benutzererfahrung auf 3G- und 4G-Netzwerken	HDX In den meisten Fällen ergeben die standardmäßigen Citrix Richtlinienereinstellungen die beste Benutzererfahrung.
Für die Fingereingabe optimierte Schnittstelle zu virtuellen Desktops und Anwendungen, optimiert für Tablet-Geräte.	Einstellungen der Richtlinie "Mobilerfahrung" Die Standardeinstellungen liefern im Allgemeinen die beste Benutzererfahrung.
Remotezugriff auf Büro-PCs von anderen Geräten, einschließlich Smartphones, Tablets, Laptops und PCs	Remote-PC-Zugriff Citrix Receiver
Automatisch bereitgestellte Anwendungen für alle Benutzer eines Stores	Citrix StoreFront Citrix App Controller
E-Mail-basierte Kontenermittlung mit benutzerseitiger Kontoeinrichtung durch Eingabe der E-Mail-Adresse	Citrix StoreFront Citrix NetScaler Gateway (für Remoteverbindungen)
Sicherer Zugriff auf Stores, Desktops und Anwendungen	Citrix StoreFront Citrix NetScaler Gateway
Vereinfachte Druckverwaltung mit Netzwerkdrucken von jedem Gerät	Universeller Druckserver
Optimierte Leistung und Bereitstellung der Dienste für Zweigstellen und mobile Benutzer	Citrix CloudBridge
Rollenbasierte Verwaltung, Konfiguration und Sicherheit für Unternehmens- und mitarbeitereigene Mobilgeräte	Mobile Solutions Bundle (XenMobile MDM und CloudGateway)

Active Directory

Sep 29, 2015

Active Directory ist zum Authentifizieren und Autorisieren erforderlich. Mit der Kerberos-Infrastruktur in Active Directory wird die Authentizität und Vertraulichkeit der Kommunikation zwischen den Delivery Controllern garantiert. Informationen zu Kerberos finden Sie in der Dokumentation von Microsoft.

Für die Gesamtstruktur und Domäne benötigen Sie eine der folgenden Funktionsebenen:

- Windows 2000 native
- Windows Server 2003
- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2012 R2

Zur Verwendung der Gruppenrichtlinienmodellierung muss der Controller auf einem Server unter Windows 2003 oder höher bis Windows Server 2012 R2 ausgeführt werden. Dies wirkt sich nicht auf die Domänenfunktionsebene aus.

Dieses Produkt unterstützt Folgendes:

- Bereitstellungen, in denen die Benutzerkonten und Computerkonten in Domänen in einer einzigen Active Directory-Gesamtstruktur bestehen. Benutzer- und Computerkonten können in beliebigen Domänen in einer Gesamtstruktur bestehen. Alle Domänen- und Gesamtstrukturebenen werden in diesem Bereitstellungstyp unterstützt.
- Bereitstellungen, in denen die Benutzerkonten und die Computerkonten der Controller und virtuellen Desktops in unterschiedlichen Active Directory-Gesamtstrukturen bestehen. Bei diesem Bereitstellungstyp muss eine Vertrauensstellung zwischen den Domänen mit den Computerkonten der Controller und virtuellen Desktops und den Domänen mit den Benutzerkonten bestehen. Sie können Gesamtstruktur- oder externe Vertrauensstellungen verwenden. Alle Domänen- und Gesamtstrukturebenen werden in diesem Bereitstellungstyp unterstützt.
- Bereitstellungen, in denen die Computerkonten für Controller in einer Active Directory-Gesamtstruktur bestehen, die sich von den zusätzlichen Active Directory-Gesamtstrukturen mit den Computerkonten für die virtuellen Desktops unterscheidet. Bei diesem Bereitstellungstyp muss eine bidirektionale Vertrauensstellung zwischen den Domänen mit den Computerkonten der Controller und allen Domänen mit den Computerkonten der virtuellen Desktops bestehen. Bei diesem Bereitstellungstyp müssen alle Domänen mit Computerkonten für Controller oder virtuelle Desktops mindestens auf der Funktionsebene "Windows 2000 native" sein. Alle Funktionsebenen der Gesamtstruktur werden unterstützt.
- Beschreibbarer Domänencontroller. Schreibgeschützte Domänencontroller werden nicht unterstützt.

Virtual Delivery Agents (VDAs) können mit in Active Directory veröffentlichten Informationen die Controller ermitteln, bei denen sie sich registrieren können (Discovery). Diese Methode wird primär für Abwärtskompatibilität unterstützt und ist nur verfügbar, wenn die VDAs und die Controller in derselben Active Directory-Gesamtstruktur sind. Informationen über diese Discovery-Methode finden Sie unter [Active Directory-basierte Controller-Discovery](#) und [CTX118976](#).

Bereitstellen in einer Active Directory-Umgebung mit mehreren Gesamtstrukturen

Aktualisiert: 2014-05-29

Hinweis: Diese Informationen gelten für Versionen ab XenDesktop 7.1 und XenApp 7.5. Sie gelten nicht für ältere Versionen von XenDesktop und XenApp.

Bei einer Active Directory-Umgebung mit mehreren Gesamtstrukturen und unidirektionalen oder bidirektionalen

Vertrauensstellungen können Sie DNS-Weiterleitungen zur Suche und Registrierung von Namen verwenden. Mit dem Assistenten zum Zuweisen der Objektverwaltung können Sie den entsprechenden Active Directory-Benutzern das Erstellen von Computerkonten ermöglichen. Weitere Informationen zu diesem Assistenten finden Sie in der Dokumentation von Microsoft.

In der DNS-Infrastruktur sind keine Reverse-DNS-Zonen erforderlich, wenn die entsprechenden DNS-Weiterleitungen zwischen Gesamtstrukturen eingerichtet sind.

In diesem Release ist der SupportMultipleForest-Schlüssel erforderlich, wenn der Virtual Delivery Agent (VDA) und der Delivery Controller in unterschiedlichen Gesamtstrukturen sind, unabhängig davon, ob sich die Active Directory- und NetBIOS-Namen voneinander unterscheiden. Der Schlüssel "SupportMultipleForest" ist nur auf dem VDA nötig. Mit den folgenden Informationen fügen Sie einen Registrierungsschlüssel hinzu:

- HKEY_LOCAL_MACHINE\Software\Citrix\VirtualDesktopAgent\SupportMultipleForest
 - Name: SupportMultipleForest
 - Typ: REG_DWORD
 - Wert: 0x00000001 (1)

Setzen Sie den Wert auf 1.

Sie müssen möglicherweise die DNS-Konfiguration umkehren, wenn sich der DNS-Namespace vom Active Directory-Namespace unterscheidet.

Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab. Wenn externe Vertrauensstellungen während des Setups vorhanden sind, ist der Registrierungsschlüssel "ListOfSIDs" erforderlich. Der Registrierungsschlüssel "ListOfSIDs" ist auch erforderlich, wenn der vollqualifizierte Domänenname (FQDN) für Active Directory sich vom DNS-FQDN unterscheidet oder die Domäne mit dem Domänencontroller einen anderen Netbios-Namen hat als der Active Directory-FQDN. Verwenden Sie zum Hinzufügen des Registrierungsschlüssels die folgenden Informationen:

- Für einen 32-Bit- oder 64-Bit-VDA verwenden Sie folgenden Registrierungsschlüssel:
HKEY_LOCAL_MACHINE\Software\Citrix\VirtualDesktopAgent\ListOfSIDs
 - Name: ListOfSIDs
 - Typ: REG_SZ
 - Daten: Sicherheits-ID (SID) der Controller

Verwenden Sie zum Abrufen Ihrer Domänen-SID ADEplorer oder XDPing.

Wenn externe Vertrauensbeziehungen vorhanden sind, müssen Sie zudem die folgenden Änderungen auf dem VDA vornehmen:

1. Navigieren Sie zur folgenden Datei: \Citrix\Virtual Desktop Agent\brokeragentconfig.exe.config
2. Erstellen Sie eine Sicherungskopie der Datei.
3. Öffnen Sie die Datei in einem Textbearbeitungsprogramm, z. B. Notepad.
4. Suchen Sie den Text allowNtlm="false" und ändern Sie den Text zu allowNtlm="true"
5. Speichern Sie die Datei.

Nachdem der Registrierungsschlüssel "ListOfSIDs" hinzugefügt wurde und die Datei brokeragent.exe.config bearbeitet wurde, müssen Sie den Citrix Desktopdienst neu starten, damit die Änderungen angewendet werden.

Die folgende Tabelle enthält die unterstützten Vertrauentypen:

Vertrauentyp	Transitivität	Richtung	In diesem Release unterstützt
Über-/untergeordnet	Transitiv	Bidirektional	Ja
Strukturstamm	Transitiv	Bidirektional	Ja
Extern	Nicht transitiv	Unidirektional oder bidirektional	Ja
Gesamtstruktur	Transitiv	Unidirektional oder bidirektional	Ja
Shortcut	Transitiv	Unidirektional oder bidirektional	Ja
Bereich	Transitiv oder nicht transitiv	Unidirektional oder bidirektional	Nein

Weitere Informationen über komplexe Active Directory-Umgebungen finden Sie unter [CTX134971](#).

Drucken

Sep 29, 2015

Bevor Sie die Bereitstellung planen, sollten Sie mit folgenden Hauptkonzepten des Druckens vertraut sein:

- Verfügbare Typen der Druckerbereitstellung
- Wie Druckaufträge weitergeleitet werden
- Grundlagen der Druckertreiberverwaltung

Die Druckkonzepte bauen auf denen von Windows auf. Um das Drucken in Ihrer Umgebung zu konfigurieren und erfolgreich zu verwalten, müssen Sie verstehen, wie das Netzwerk- und Clientdrucken in Windows funktioniert und wie das Druckverhalten in dieser Umgebung umgesetzt wird.

Ablauf des Druckprozesses

In dieser Umgebung werden alle Druckvorgänge (durch den Benutzer) auf Maschinen initiiert, auf denen Anwendungen gehostet werden. Druckaufträge werden über den Netzwerkdruckserver oder das Benutzergerät an das Druckgerät weitergeleitet.

Für Benutzer von virtuellen Desktops und Anwendungen gibt es keinen persistenten Arbeitsbereich. Bei Sitzungsende wird der Arbeitsbereich des Benutzers gelöscht, demnach müssen alle Einstellungen zu Beginn jeder Sitzung neu erstellt werden. Bei jedem Start einer neuen Sitzung muss daher die Neuerstellung des Arbeitsbereichs durch das System erfolgen.

Wenn ein Benutzer druckt, übernimmt das System folgende Aufgaben:

- Entscheidung darüber, welche Drucker dem Benutzer bereitgestellt werden. Dies wird als Druckerprovisioning bezeichnet.
- Wiederherstellen der Druckereinstellungen des Benutzers.
- Ermitteln des Standarddruckers für die Sitzung.

Sie können festlegen, wie diese Aufgaben durchgeführt werden, indem Sie die Optionen für das Druckerprovisioning, die Weiterleitung von Druckaufträgen, das Speichern von Druckereigenschaften und die Treiberverwaltung konfigurieren. Bedenken Sie dabei, wie die verschiedenen Einstellungen möglicherweise die Druckleistung in der Umgebung und die Benutzererfahrung beeinflussen.

Druckerprovisioning

Der Prozess, durch den Drucker in einer Sitzung verfügbar gemacht werden, wird als Provisioning bezeichnet. Das Druckerprovisioning wird normalerweise dynamisch abgewickelt. Das heißt, die in einer Sitzung angezeigten Drucker sind nicht vordefiniert und gespeichert. Stattdessen werden die Drucker gemäß der Richtlinien beim Entstehen der Sitzung während der Anmeldung und Wiederverbindung zusammengestellt. Folglich können sich die Drucker je nach Richtlinie, Benutzerort und Netzwerkänderungen ändern, vorausgesetzt, dies spiegelt sich in den Richtlinien wider. Benutzer, die an einen anderen Ort wechseln, bemerken daher möglicherweise Änderungen in ihrem Arbeitsbereich.

Das System überwacht auch clientseitige Drucker und passt automatisch erstellte Drucker in Sitzungen dynamisch an, je nachdem, welche Hinzufügungen, Löschungen und Änderungen an den clientseitigen Druckern vorgenommen werden. Von dieser dynamischen Druckerermittlung profitieren mobile Benutzer, wenn sie über verschiedene Geräte eine Verbindung herstellen.

In diesem Abschnitt werden die gängigsten Methoden des Druckerprovisioning vorgestellt.

Universeller Druckserver

Der universelle Druckserver von Citrix bietet universelle Druckunterstützung für Netzwerkdrucker. Der universelle Druckserver verwendet den universellen Druckertreiber. Diese Lösung ermöglicht die Verwendung eines einzelnen Treibers auf einer Serverbetriebssystemmaschine und damit den Netzwerkdruck von jedem Gerät aus.

Citrix empfiehlt den Einsatz des universellen Druckservers für Szenarios mit Remote-Druckerservern. Der universelle Druckserver überträgt den Druckauftrag über das Netzwerk in einem optimierten und komprimierten Format, wodurch der Netzwerkverkehr reduziert und die Benutzererfahrung verbessert wird.

Der universelle Druckserver umfasst als Feature die folgenden Komponenten:

- Eine Clientkomponente namens "UPClient".

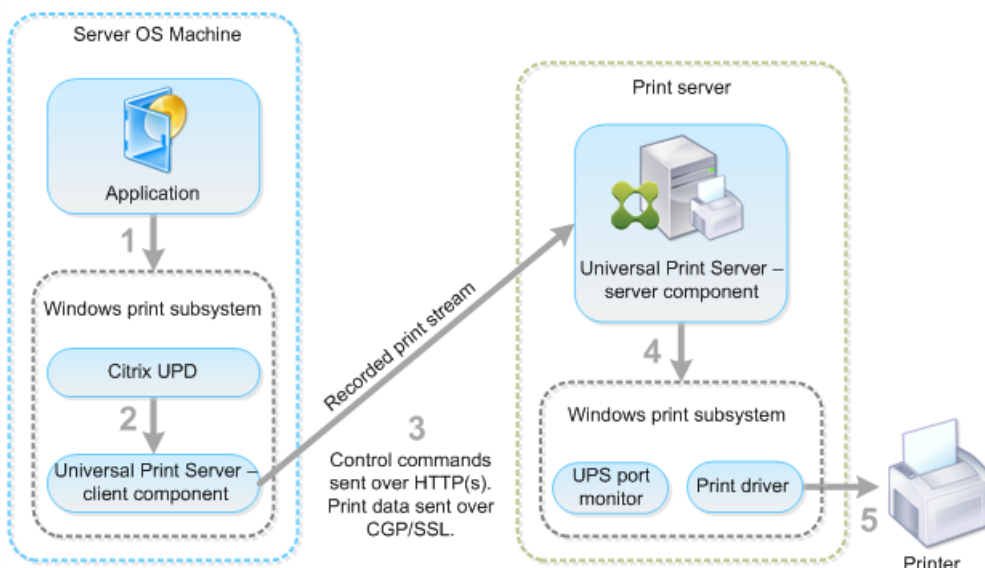
Aktivieren Sie UPClient auf jeder Serverbetriebssystemmaschine, die Netzwerkdrucker bereitstellt und den universellen Druckertreiber verwendet.

- Eine Serverkomponente namens "UPServer".

Installieren Sie UPServer auf jedem Druckserver, der Sitzungs-Netzwerkdrucker bereitstellt und den universellen Druckertreiber für die Sitzungsdrucker verwendet (unabhängig davon, ob die Sitzungsdrucker zentral bereitgestellt werden).

Informationen zu den Anforderungen und zum Setup des universellen Druckservers finden Sie in den Abschnitten über Systemanforderungen und Installation.

Hinweis: Der universelle Druckserver wird auch für VDI-in-a-Box 5.3 unterstützt. Informationen zur Installation des universellen Druckservers mit VDI-in-a-Box finden Sie in den Abschnitten zur Installation von VDI-in-a-Box in den eDocs. Die folgende Abbildung zeigt den typischen Workflow eines Netzwerkdruckers in einer Umgebung mit universellem Druckserver.



Wenn Sie das Citrix Feature "Universeller Druckserver" aktivieren, wird es von allen verbundenen Netzwerkdruckern automatisch über Autodiscovery genutzt.

Automatische Erstellung

Automatische Erstellung bezieht sich auf Drucker, die automatisch zu Beginn jeder Sitzung erstellt werden. Sowohl

Remotedrucker als auch lokal angeschlossene Drucker können automatisch erstellt werden. Bei Umgebungen mit einer großen Anzahl von Druckern pro Benutzer ist es u. U. besser, nur den Standarddrucker automatisch zu erstellen. Wenn weniger Drucker automatisch erstellt werden, entsteht auf den Serverbetriebssystemmaschinen weniger Mehraufwand (Arbeitsspeicher und CPU). Eine reduzierte Anzahl an automatisch erstellten Druckern kann auch die Anmeldedauer der Benutzer verkürzen.

Automatisch erstellte Drucker basieren auf:

- Den auf dem Benutzergerät installierten Druckern.
- Den auf die Sitzung angewendeten Richtlinien.
Durch RichtlinienEinstellungen für die automatische Erstellung können Sie Anzahl oder Art der automatisch erstellten Drucker beschränken. Standardmäßig sind die Drucker in Sitzungen verfügbar, wenn alle Drucker auf dem Benutzergerät automatisch konfiguriert werden, einschließlich der lokal angeschlossenen und der Netzwerkdrucker.

Nachdem der Benutzer die Sitzung beendet, werden die Drucker für diese Sitzung gelöscht.

Mit der automatischen Erstellung von Client- und Netzwerkdruckern sind Wartungsarbeiten verbunden. Bei Hinzufügen eines Druckers muss beispielsweise auch Folgendes durchgeführt werden:

- Aktualisieren der RichtlinienEinstellung Sitzungsdrucker
- Hinzufügen des Treibers zu allen Serverbetriebssystemmaschinen über die RichtlinienEinstellung Druckertreiberzuordnung und -kompatibilität

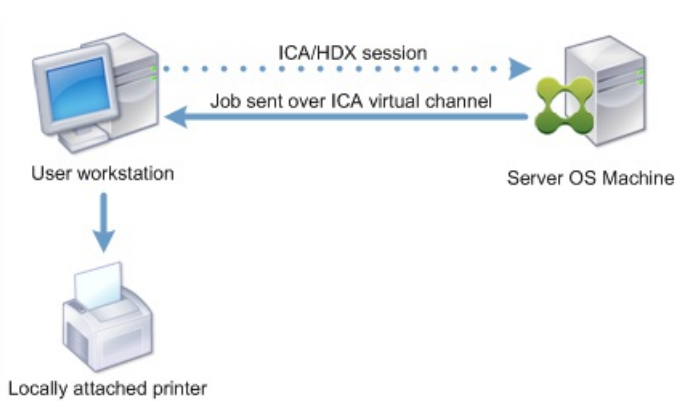
Weiterleiten von Druckaufträgen

Der Begriff Druckpfad umfasst den Pfad, über den Druckaufträge weitergeleitet werden, und den Speicherort, an dem Druckaufträge gespoolt werden. Beide Aspekte dieses Konzepts sind wichtig. Die Weiterleitung wirkt sich auf den Netzwerk-Datenverkehr aus. Das Spooling wirkt sich auf die Auslastung der lokalen Ressourcen an dem Gerät, das den Auftrag verarbeitet, aus.

In dieser Umgebung können Druckaufträge auf zwei Wegen zu einem Druckgerät gelangen: über den Client oder über einen Netzwerkdruckserver. Dafür werden die Bezeichnungen Clientdruckpfad und Netzwerkdruckpfad verwendet. Welcher Pfad standardmäßig ausgewählt wird, hängt vom verwendeten Drucker ab.

Lokal angeschlossene Drucker

Das System leitet Aufträge von der Serverbetriebssystemmaschine über den Client an den Drucker. Der Druckdatenverkehr wird über das ICA-Protokoll optimiert und komprimiert. Wenn ein Druckgerät lokal an das Benutzergerät angeschlossen ist, werden Druckaufträge über den virtuellen ICA-Kanal weitergeleitet.



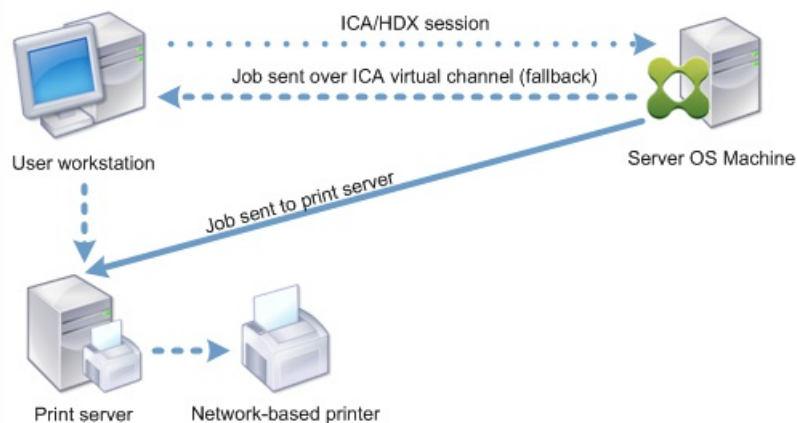
Netzwerkdrucker

Standardmäßig werden alle für Netzwerkdrucker bestimmten Druckaufträge von der Serverbetriebssystemmaschine über das Netzwerk direkt an den Druckserver weitergeleitet. In folgenden Situationen werden jedoch Druckaufträge automatisch über die ICA-Verbindung geleitet:

- Wenn der virtuelle Desktop oder die Anwendung keine Verbindung mit dem Druckserver herstellen kann.
- Wenn der systemeigene Druckertreiber auf der Serverbetriebssystemmaschine nicht verfügbar ist.

Wenn der universelle Druckserver nicht aktiviert ist, empfiehlt sich die Konfiguration des Clientdruckpfads für den Netzwerkdruck bei Verbindungen mit geringer Bandbreite, z. B. WANs, die von der Optimierung und Komprimierung des Datenverkehrs beim Senden von Aufträgen über die ICA-Verbindung profitieren.

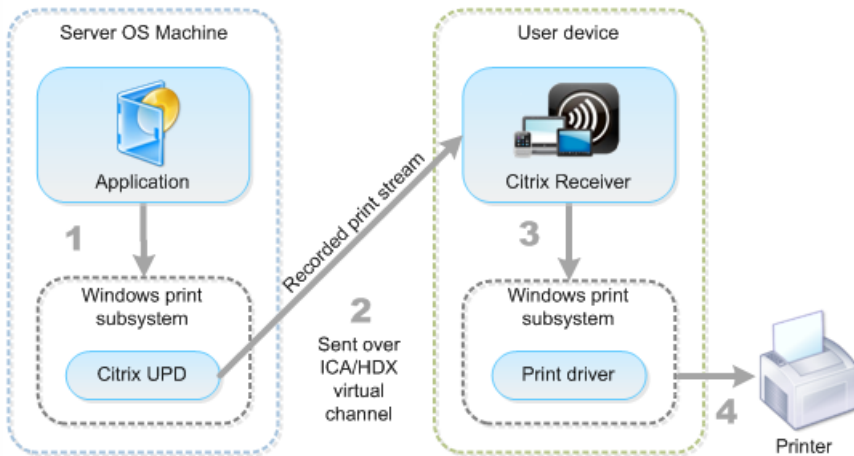
Der Clientdruckpfad ermöglicht auch die Begrenzung des Datenverkehrs oder der für Druckaufträge zugeordneten Bandbreite. Wenn die Auftragsleitung über das Benutzergerät nicht möglich ist, z. B. bei Thin Clients ohne Druckerfunktionen, muss die Servicequalität so konfiguriert werden, dass ICA/HDX-Verkehr Vorrang hat und eine gute Benutzererfahrung bei der Sitzung gewährleistet ist.



Druckertreiberverwaltung

Zur Vereinfachung der Druckprozesse in diesen Umgebungen empfiehlt Citrix die Verwendung des universellen Citrix Drucktreibers. Der universelle Druckertreiber ist ein geräteunabhängiger, für jedes Druckgerät geeigneter Treiber. Durch die so erzielte Reduzierung der Treiberanzahl wird die Verwaltung erheblich vereinfacht. Der universelle Druckertreiber unterstützt die erweiterten Druckerfunktionen, wie z. B. Heftung und Sortierung und beschränkt nicht die Farbtiefe.

Die folgende Abbildung zeigt die universellen Druckertreiberkomponenten und einen typischen Workflow für ein lokal angeschlossenes Druckgerät.

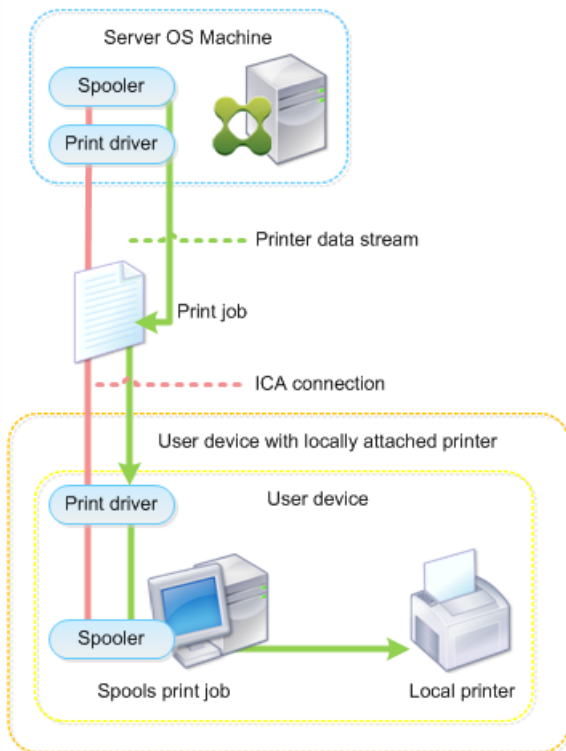


Legen Sie bei der Planung der Strategie zur Treiberverwaltung fest, ob Sie gerätespezifische Treiber, den universellen Druckertreiber oder beides unterstützen wollen. Wenn Sie Standardtreiber unterstützen, müssen Sie außerdem Folgendes festlegen:

- Zu unterstützende Treibertypen
- Aktivieren oder Deaktivieren der automatischen Installation der Druckertreiber (falls auf Serverbetriebssystemmaschinen nicht vorhanden)
- Erstellen der Treiberkompatibilitätslisten

Wenn das System während der automatischen Druckererstellung erkennt, dass ein neuer lokaler Drucker an einem Benutzergerät angeschlossen ist, wird die Serverbetriebssystemmaschine auf den erforderlichen Druckertreiber hin überprüft. Ist kein Windows-systemeigener Treiber verfügbar, wird vom System standardmäßig der universelle Druckertreiber verwendet.

Der Druckvorgang kann nur dann erfolgreich ausgeführt werden, wenn der Druckertreiber auf der Serverbetriebssystemmaschine und der Treiber auf dem Benutzergerät übereinstimmen. In der folgenden Abbildung wird dargestellt, wie der Druckertreiber an zwei Orten für den Clientdruck verwendet wird.



Bewährte Methoden

Aktualisiert: 2015-04-26

Viele Faktoren bestimmen die beste Drucklösung für eine bestimmte Umgebung. Einige dieser bewährten Methoden sind möglicherweise für Ihre Umgebung nicht geeignet.

- Verwenden Sie das Citrix Feature "universeller Druckserver".
- Verwenden Sie den universellen Druckertreiber oder Windows-systemeigene Treiber.
- Reduzieren Sie die Anzahl der installierten Druckertreiber auf den Serverbetriebssystemmaschinen auf das Minimum.
- Verwenden Sie Treiberzuordnung zu systemeigenen Treibern.
- Installieren Sie nie ungetestete Druckertreiber in einer Produktionssite.
- Vermeiden Sie Updates von Treibern. Versuchen Sie stets, einen Treiber zu deinstallieren, den Server neu zu starten und dann einen Ersatztreiber zu installieren.
- Deinstallieren Sie nicht verwendete Treiber oder verwenden Sie die Richtlinie Druckertreiberzuordnung und -kompatibilität, um zu verhindern, dass Drucker mit dem Treiber erstellt werden.
- Vermeiden Sie möglichst Kernelmodustreiber der Version 2.
- Wenden Sie sich an den Hersteller oder sehen Sie in der Citrix Ready-Produktdokumentation www.citrix.com/ready nach, ob ein Druckermodell unterstützt wird.

Im Allgemeinen werden alle von Microsoft zur Verfügung gestellten Druckertreiber mit Terminaldiensten getestet und ihre Funktion unter Citrix gewährleistet. Vergewissern Sie sich jedoch vor Einsatz eines Druckertreibers eines Drittanbieters, dass der Treiber von Windows Hardware Quality Labs (WHQL) für Terminaldienste zertifiziert wurde. Citrix vergibt keine Zertifizierung für Druckertreiber.

Überlegungen zur Sicherheit beim Drucken

Citrix Drucklösungen sind inhärent sicher.

- Der Citrix Druckmanagerdienst überwacht und reagiert fortlaufend auf Sitzungsereignisse wie An- und Abmeldung,

Trennen, Wiederverbinden und Beenden der Sitzung. Er behandelt Anforderungen, indem er die Identität des Benutzers der aktuellen Sitzung übernimmt.

- Beim Citrix Drucken wird jedem Drucker ein eindeutiger Namespace in einer Sitzung zugewiesen.
- Die Standardsicherheitsbeschreibung für automatisch erstellte Drucker wird eingerichtet, um sicherzustellen, dass die in einer Sitzung automatisch erstellten Clientdrucker für Benutzer in anderen Sitzungen nicht zugänglich sind. Standardmäßig können Administratoren nicht versehentlich auf einem Clientdrucker einer anderen Sitzung drucken, obwohl sie jeden Clientdrucker sehen und die Berechtigungen dafür manuell ändern können.

Druckrichtlinien und Einstellungen

Aktualisiert: 2013-06-18

Wenn Benutzer von veröffentlichten Anwendungen aus auf Drucker zugreifen, können Sie über Citrix Richtlinien Folgendes konfigurieren:

- Wie das Drucker-Provisioning erfolgt (bzw. wie Drucker zu Sitzungen hinzugefügt werden)
- Wie Druckaufträge weitergeleitet werden
- Wie Druckertreiber verwaltet werden

Sie können verschiedene Druckkonfigurationen für unterschiedliche Benutzergeräte, Benutzer oder beliebige andere Objekte haben, nach denen Richtlinien gefiltert werden.

Die meisten Druckfunktionen werden über die Citrix Druckrichtlinien konfiguriert. Druckeinstellungen folgen dem Standardverhalten für Citrix Richtlinien:

Druckereinstellungen können vom System am Ende einer Sitzung in das Druckerobjekt oder das Clientdruckgerät geschrieben werden, sofern das Netzwerkkonto des Benutzers ausreichende Berechtigungen hat. Standardmäßig verwendet Receiver die Einstellungen, die im Druckerobjekt in der Sitzung gespeichert wurden, bevor an anderen Orten nach Einstellungen gesucht wird.

Standardmäßig werden die Druckereigenschaften auf dem Benutzergerät (falls vom Gerät unterstützt) oder im Benutzerprofil auf der Serverbetriebssystemmaschine gespeichert oder beibehalten. Wenn die Druckereigenschaften während einer Sitzung vom Benutzer geändert werden, werden diese Änderungen im Benutzerprofil auf der Maschine aktualisiert. Wenn sich der Benutzer das nächste Mal anmeldet oder eine neue Verbindung herstellt, übernimmt das Benutzergerät die beibehaltenen Einstellungen. Das heißt, auf dem Benutzergerät geänderte Druckereigenschaften wirken sich nicht auf die aktuelle Sitzung aus bis zum Ab- und erneuten Anmelden des Benutzers.

Druckeinstellungen

Allgemeine Speicherorte der Druckeinstellungen

In Windows-Druckumgebungen können die an den Druckvoreinstellungen vorgenommenen Änderungen auf dem lokalen Computer oder in einem Dokument gespeichert werden. Wenn Benutzer in dieser Umgebung Druckeinstellungen ändern, können diese Änderungen an folgenden Positionen gespeichert werden:

- **Auf dem Benutzergerät:** Windows-Benutzer können Geräteeinstellungen auf dem Benutzergerät ändern, indem sie mit der rechten Maustaste auf die Drucker in der Systemsteuerung klicken und Druckeinstellungen wählen. Wenn beispielsweise Querformat als Seitenausrichtung ausgewählt wird, gilt Querformat als Standard-Seitenausrichtung für diesen Drucker.
- **In einem Dokument:** Bei Textverarbeitungs- und Desktop-Publishing-Programmen werden Dokumenteinstellungen, z. B. die Seitenausrichtung, häufig in Dokumenten gespeichert. Wenn Sie beispielsweise ein zu druckendes Dokument in eine

Warteschlange setzen, speichert Microsoft Word die von Ihnen angegebenen Druckvoreinstellungen wie Seitenausrichtung und Druckernamen im Dokument selbst. Diese Einstellungen erscheinen standardmäßig, wenn Sie dieses Dokument das nächste Mal drucken.

- **Benutzerseitige Änderungen in einer Sitzung:** Das System übernimmt Änderungen an den Druckereinstellungen eines automatisch erstellten Druckers nur, wenn diese in der Systemsteuerung der Sitzung vorgenommen wurden, also auf der Serverbetriebssystemmaschine.
- **Auf der Serverbetriebssystemmaschine:** Dies sind die Standardeinstellungen, die einem bestimmten Druckertreiber auf der Maschine zugeordnet sind.

Die in einer Windows-Umgebung gespeicherten Einstellungen sind abhängig von der Stelle, an der die Einstellungen vom Benutzer vorgenommen wurden. Das bedeutet außerdem, dass die an einer Stelle wie einer Tabellenkalkulation angezeigten Druckereinstellungen sich von den Einstellungen an anderen Stellen, beispielsweise in Dokumenten, unterscheiden können. Die auf einen bestimmten Drucker angewendeten Druckereinstellungen variieren daher innerhalb einer Sitzung.

Hierarchie der Benutzerdruckereinstellungen

Da die Druckereinstellungen an verschiedenen Stellen gespeichert werden können, verarbeitet das System sie gemäß einer bestimmten Priorität. Sie dürfen auch nicht vergessen, dass Geräteeinstellungen anders behandelt werden als Dokumenteinstellungen und normalerweise Vorrang vor diesen haben.

Standardmäßig wendet das System immer alle Druckereinstellungen an, die ein Benutzer während einer Sitzung geändert hat, d. h. alle beibehaltenen Einstellungen, bevor andere Einstellungen berücksichtigt werden. Wenn der Benutzer druckt, führt das System die auf der Serverbetriebssystemmaschine gespeicherten Standarddruckereinstellungen mit allen beibehaltenen Einstellungen oder Clientdruckereinstellungen zusammen und wendet sie an.

Speichern der Druckereinstellungen des Benutzers

Citrix empfiehlt, dass Sie den Speicherort der Druckereigenschaften nicht ändern. Am einfachsten können Sie konsistente Druckereigenschaften sicherstellen, indem Sie die Standardeinstellung beibehalten, wonach die Druckereigenschaften auf dem Benutzergerät gespeichert werden. Wenn das System die Eigenschaften auf dem Benutzergerät nicht speichern kann, wird automatisch auf das Benutzerprofil auf der Serverbetriebssystemmaschine zurückgegriffen.

Überprüfen Sie die Richtlinieneinstellung Speicherung von Druckereigenschaften, wenn diese Szenarios zutreffen:

- Verwendung von älteren Plug-Ins, durch die das Speichern der Druckereigenschaften durch die Benutzer auf einem Benutzergerät unterbunden wird
- Verwendung verbindlicher Profile im Windows-Netzwerk, wobei die Druckereigenschaften der Benutzer beibehalten werden sollen

Standarddruckvorgänge

Aktualisiert: 2013-06-18

Wenn Sie keine Richtlinienregeln konfigurieren, zeigt sich standardmäßig das folgende Druckverhalten:

- Universeller Druckserver ist deaktiviert.
- Alle auf dem Benutzergerät konfigurierten Drucker werden automatisch zu Beginn jeder Sitzung konfiguriert. Dieses Verhalten entspricht der Richtlinieneinstellung Clientdrucker automatisch erstellen mit der Option Alle Clientdrucker automatisch erstellen.
- Das System leitet alle Druckaufträge, die in Warteschlangen für an Benutzergeräte lokal angeschlossene Drucker

gestellt wurden, als Clientdruckaufträge weiter (d. h. über den ICA-Kanal und durch das Benutzergerät).

- Das System leitet alle Druckaufträge, die in Warteschlangen von Netzwerkdruckern gestellt wurden, direkt über Serverbetriebssystemmaschinen. Falls die Aufträge vom System nicht über das Netzwerk weitergeleitet werden können, werden sie als umgeleiteter Clientdruckauftrag über das Benutzergerät weitergeleitet.
Dieses Verhalten entspricht dem Deaktivieren der Citrix Richtlinieneinstellung Direkte Verbindungen zu Druckservern.
- Standardmäßig versucht das System, die Druckeigenschaften (eine Kombination aus den Druckeinstellungen des Benutzers und den gerätespezifischen Druckeinstellungen) auf dem Benutzergerät zu speichern. Wenn der Client diesen Vorgang nicht unterstützt, werden die Druckeigenschaften vom System in Benutzerprofilen auf der Serverbetriebssystemmaschine gespeichert.
Dieses Verhalten entspricht der Citrix Richtlinieneinstellung Speicherung von Druckereigenschaften mit der Option Nur im Profil speichern, wenn sie nicht auf dem Client gespeichert sind.
- Das System verwendet die Windows-Version des Druckertreibers, falls sie auf der Serverbetriebssystemmaschine verfügbar ist. Ist der Druckertreiber nicht verfügbar, versucht das System, den Treiber vom Windows-Betriebssystem zu installieren. Ist der Treiber in Windows nicht verfügbar, wird ein universeller Citrix Druckertreiber verwendet.
Dieses Verhalten entspricht dem Aktivieren der Citrix Richtlinieneinstellung Automatische Installation von mitgelieferten Druckertreibern und Konfigurieren der Einstellung Universal Printing mit der Option Universelles Drucken nur verwenden, wenn angeforderter Treiber nicht verfügbar ist.

Das Aktivieren von Automatische Installation von mitgelieferten Druckertreibern kann dazu führen, dass eine große Anzahl systemeigener Druckertreiber installiert wird.

Hinweis: Wenn Sie nicht sicher sind, welche Standardwerte voreingestellt sind, zeigen Sie sie an, indem Sie eine neue Richtlinie erstellen und alle Druckrichtlinienregeln aktivieren. Die angezeigte Option ist die Standardoption.

Beispiel einer Druckkonfiguration

Aktualisiert: 2013-06-18

Die Auswahl der am besten geeigneten Druckkonfigurationsoptionen für die Anforderungen und die Umgebung kann die Verwaltung vereinfachen. Obwohl die Standarddruckkonfiguration für die meisten Umgebungen geeignet ist, gewährleisten die Standardwerte möglicherweise nicht die erwartete Benutzererfahrung oder die optimale Netzwerkverwendung und den gewünschten Verwaltungsaufwand für die Umgebung.

Die Druckkonfiguration hängt von folgenden Faktoren ab:

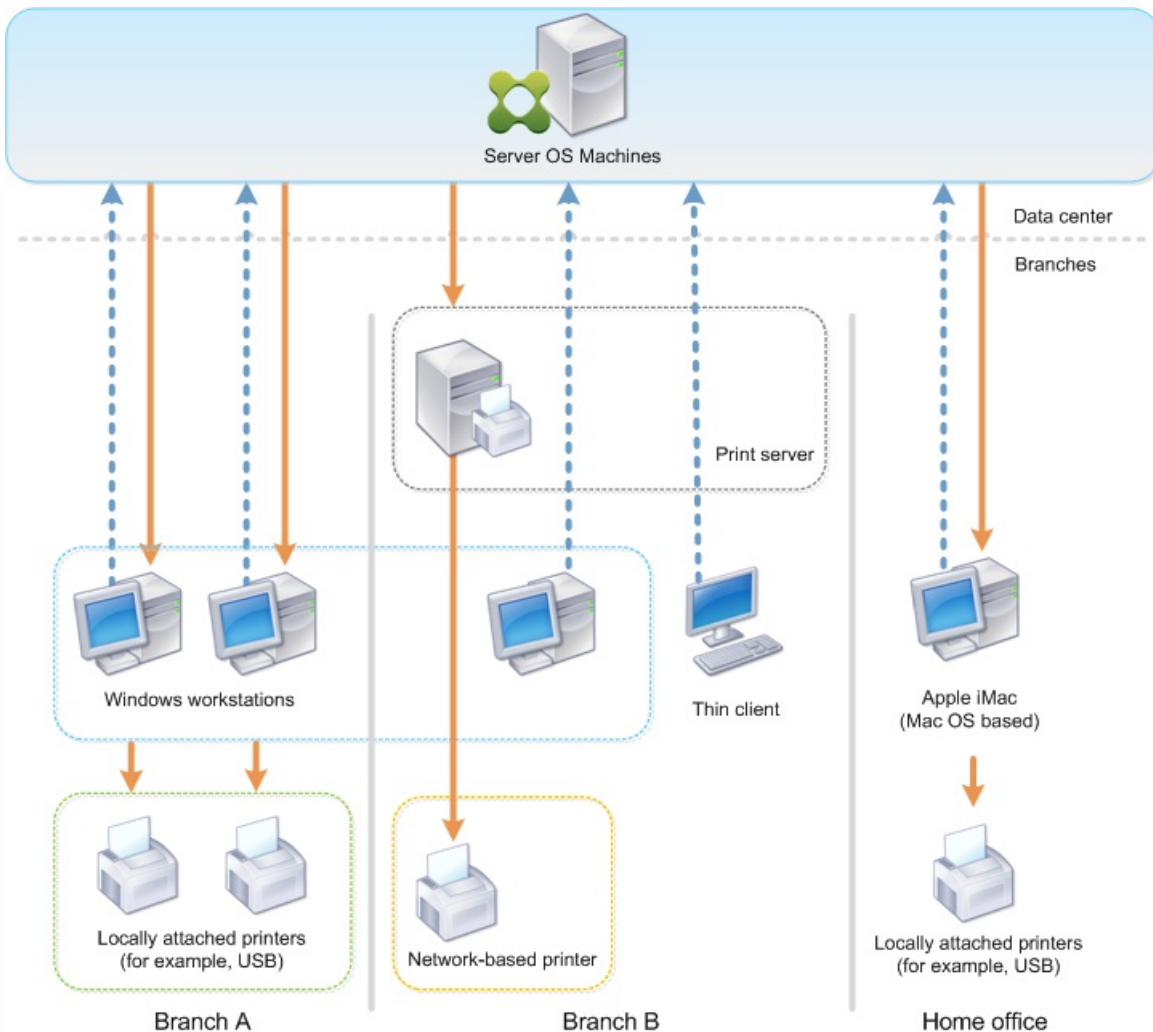
- Den Unternehmensanforderungen und der vorhandenen Druckinfrastruktur.
Berücksichtigen Sie bei der Druckkonfiguration die Anforderungen der Organisation. Die vorhandene Druckimplementierung (ob Benutzer Drucker hinzufügen können, welche Benutzer Zugriff auf welche Drucker haben usw.) kann bei der Definition der Druckkonfiguration ein nützlicher Leitfaden sein.
- Ob in Ihrer Organisation Sicherheitsrichtlinien gelten, die Drucker für bestimmte Benutzer reservieren (z. B. Drucker für die Personalabteilung oder die Gehaltsabrechnung).
- Ob Benutzer drucken müssen, wenn sie nicht an ihrem primären Arbeitsort sind, z. B. Mitarbeiter, die verschiedene Arbeitsstationen verwenden oder auf Geschäftsreisen gehen.

Achten Sie beim Entwerfen der Druckkonfiguration darauf, den Benutzern die gleiche Erfahrung in einer Sitzung zu bieten, wie sie es beim Drucken von lokalen Benutzergeräten aus gewohnt sind.

Beispiel einer Druckbereitstellung

Die folgende Abbildung zeigt die Bereitstellung dieser Anwendungsfälle:

- **Zweigstelle A:** Kleine Auslandsniederlassung mit einigen Windows-Arbeitsstationen. Jede Benutzerarbeitsstation hat einen lokal angeschlossenen, privaten Drucker.
- **Zweigstelle B:** Großes Zweigstellenbüro mit Thin Clients und auf Windows basierenden Arbeitsstationen. Aus Effizienzgründen teilen sich die Benutzer dieser Zweigstelle die Netzwerkdrucker (einen pro Stockwerk). Die Druckwarteschlangen werden über Windows-Druckserver der Zweigstelle gesteuert.
- **Home Office:** Büro im Haus eines Mitarbeiters mit einem Mac OS-basierten Benutzergerät, über das auf die Citrix Infrastruktur des Unternehmens zugegriffen wird. Das Benutzergerät hat einen lokal angeschlossenen Drucker.



In den folgenden Abschnitten werden Konfigurationen beschrieben, die die Komplexität der Umgebung minimieren und die Verwaltung vereinfachen.

Automatisch erstellte Clientdrucker und der universelle Citrix Druckertreiber

In Branch A arbeiten alle Benutzer auf Arbeitsstationen unter Windows und verwenden daher automatisch erstellte Clientdrucker und den universellen Druckertreiber. Dies bietet folgende Vorteile:

- **Leistung:** Druckaufträge werden über den ICA-Druckkanal geleitet, sodass die Druckdaten komprimiert werden können und Bandbreite eingespart wird.
Um sicherzustellen, dass ein einzelner Benutzer durch den Druck eines großen Dokuments nicht die Sitzungsleistung anderer Benutzer beeinträchtigt, wird eine Citrix Richtlinie für die maximale Druckbandbreite konfiguriert.

Eine andere Lösung wäre die Multistream-ICA-Verbindung, bei der der Druckverkehr innerhalb einer separaten TCP-Verbindung mit niedriger Priorität übertragen wird. Multistream-ICA kann verwendet werden, wenn Quality of Service (QoS) über die WAN-Verbindung nicht implementiert ist.

- Flexibilität: Der universelle Citrix Druckertreiber gewährleistet, dass alle mit dem Client verbundenen Drucker auch von virtuellen Desktop- oder Anwendungssitzungen verwendet werden können, ohne dass ein neuer Druckertreiber im Datacenter integriert werden muss.

Universeller Citrix Druckserver

In Branch B werden alle Netzwerkdrucker und ihre Warteschlangen auf einem Windows-Druckerserver verwaltet. Somit erweist sich der universelle Citrix Druckserver als die effizienteste Konfiguration.

Alle erforderlichen Druckertreiber werden von lokalen Administratoren auf dem Druckserver installiert und verwaltet. Das Zuordnen von Druckern in virtuellen Desktop- oder Anwendungssitzungen funktioniert wie folgt:

- Arbeitsstationen unter Windows: Das IT-Team vor Ort hilft den Benutzern beim Herstellen der Verbindung mit dem geeigneten Netzwerkdrucker auf ihren Windows-Arbeitsstationen. Dies ermöglicht Benutzern, über lokal installierte Anwendungen zu drucken.

Bei virtuellen Desktop- oder Anwendungssitzungen werden die lokal konfigurierten Drucker über die automatische Erstellung aufgelistet. Der virtuelle Desktop oder die virtuelle Anwendung stellt dann eine Verbindung mit dem Druckserver her, falls möglich, als Direktnetzwerkverbindung.

Die Komponenten des universellen Citrix Druckservers werden installiert und aktiviert, systemeigene Druckertreiber sind nicht erforderlich. Falls ein Treiber aktualisiert oder eine Druckerwarteschlange geändert wird, ist im Datacenter keine weitere Konfiguration nötig.

- Thin Clients: Für Thin Clients-Benutzer müssen die Drucker in den virtuellen Desktop- oder Anwendungssitzungen angeschlossen werden. Um den Benutzern das Drucken so einfach wie möglich zu machen, konfigurieren die Administratoren eine einzige Citrix Sitzungsdruckerrichtlinie pro Stockwerk, damit der jeweilige Drucker als Standarddrucker festgelegt wird.
Damit sichergestellt ist, dass die Benutzer stets mit dem richtigen Drucker verbunden sind, auch wenn sie in einem anderen Stockwerk sind, werden die Richtlinien nach Subnetz oder Thin Client-Namen gefiltert. Diese Konfiguration, die auch als "Proximitydrucken" bezeichnet wird, lässt die Wartung lokaler Druckertreiber zu (gemäß der delegierten Administration).

Wenn eine Druckerwarteschlange geändert oder hinzugefügt werden muss, müssen Citrix Administratoren die entsprechende Richtlinie für Sitzungsdrucker in der Umgebung ändern.

Da der Netzwerkdruckdatenverkehr außerhalb des virtuellen ICA-Kanals gesendet wird, wird QoS implementiert. Eingehende und ausgehende Netzwerkdaten an Ports für ICA/HDX-Datenverkehr haben Vorrang vor sonstigem Netzwerkdatenverkehr. Diese Konfiguration gewährleistet, dass Benutzersitzungen von großen Druckaufträgen nicht beeinträchtigt werden.

Automatisch erstellte Clientdrucker und der universelle Citrix Druckertreiber

Bei Heimbüros mit nicht standardmäßigen Arbeitsstationen und nicht verwalteten Druckgeräten ist es am einfachsten, automatisch erstellte Drucker und den universellen Druckertreiber zu verwenden.

Zusammenfassung der Bereitstellung

Zusammenfassend lässt sich die Konfiguration dieses Bereitstellungsbeispiels wie folgt beschreiben:

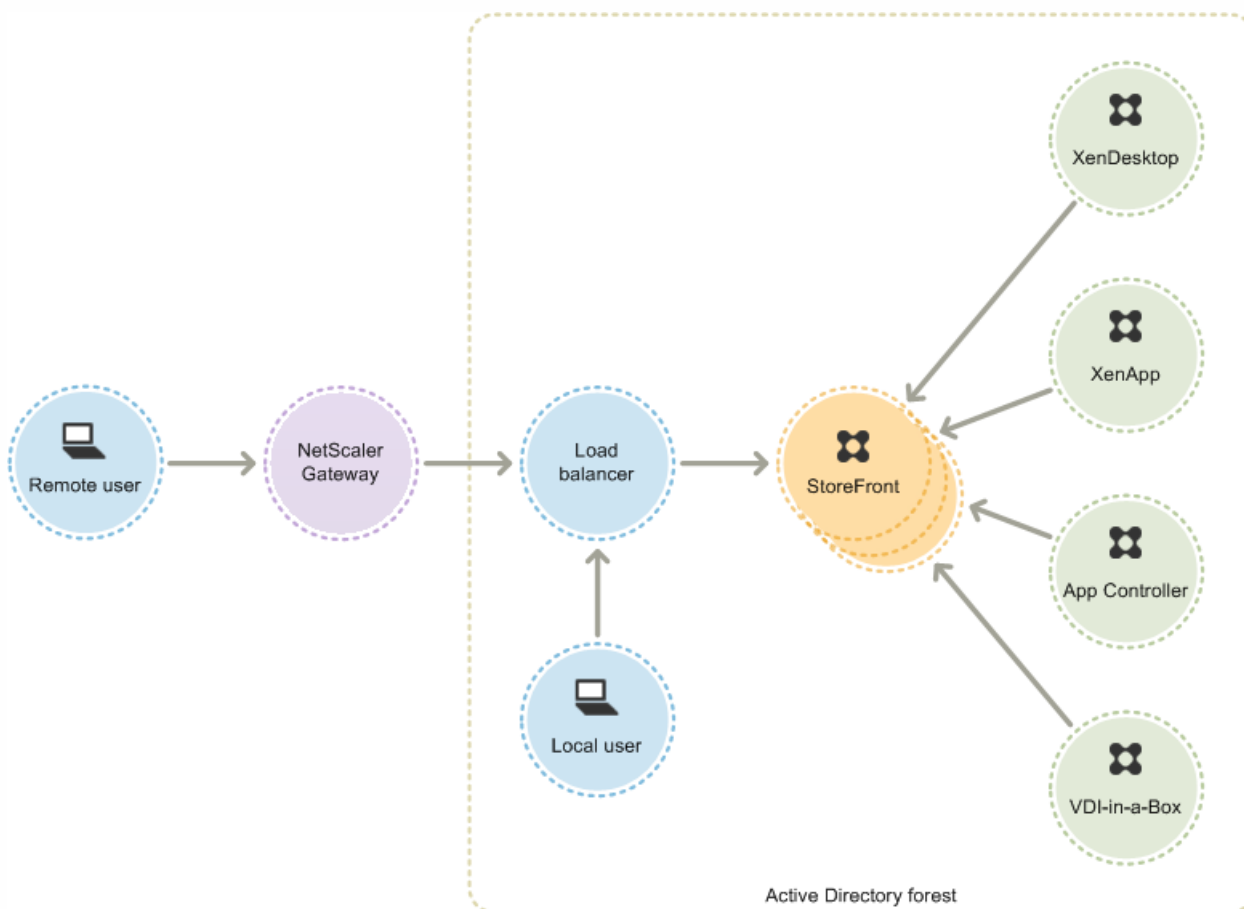
- Auf Serverbetriebssystemmaschinen werden keine Druckertreiber installiert. Es wird nur der universelle Citrix Druckertreiber verwendet. Fallback auf systemeigene Druckertreiber und die automatische Installation von Druckertreibern sind deaktiviert.
- Die automatische Erstellung von Clientdruckern für alle Benutzer wird über eine Richtlinie konfiguriert. Serverbetriebssystemmaschinen werden standardmäßig direkt mit dem Druckserver verbunden. Zur Konfiguration müssen lediglich die Komponenten des universellen Druckservers aktiviert werden.
- Eine Sitzungsdruckerrichtlinie wird für jedes Stockwerk von Branch B konfiguriert und gilt für alle Thin Clients des jeweiligen Stockwerks.
- Die Implementierung von QoS für Branch B gewährleistet eine hervorragende Benutzererfahrung.

StoreFront

Sep 29, 2015

Citrix StoreFront ist auf dem XenDesktop-Medium enthalten. Berücksichtigen Sie die folgenden Empfehlungen, wenn Sie die StoreFront-Bereitstellung planen. Weitere Informationen über StoreFront finden Sie in der [StoreFront](#)-Dokumentation in den Citrix eDocs.

- Citrix empfiehlt, StoreFront auf einer dedizierten Instanz von IIS zu hosten. Wenn andere Webanwendungen auf derselben IIS-Instanz wie StoreFront installiert werden, kann dies zu Sicherheitsproblemen für die gesamte StoreFront-Infrastruktur führen.
- In einer Produktionsumgebung empfiehlt Citrix die Verwendung von HTTPS, um den Datenverkehr zwischen StoreFront und Benutzergeräten sicher zu gestalten. Damit HTTPS verwendet werden kann, müssen die IIS-Instanz, auf der der Authentifizierungsdienst gehostet wird, und zugeordnete Stores für HTTPS konfiguriert sein. Wenn die entsprechende IIS-Konfiguration nicht verfügbar ist, verwendet StoreFront HTTP für die Kommunikation.
- StoreFront-Server müssen in derselben Microsoft Active Directory-Gesamtstruktur wie die XenDesktop- bzw. XenApp-Server sein, auf denen die Benutzerressourcen gehostet werden. Alle StoreFront-Server einer Gruppe müssen in der gleichen Domäne sein. Um die Authentifizierung mit Smartcards und Benutzerzertifikaten zu aktivieren, müssen die Benutzerkonten in der Active Directory-Gesamtstruktur konfiguriert werden, die den StoreFront-Server enthält.
- Um bei Ausfall des Hauptservers, auf dem StoreFront gehostet wird, hohe Verfügbarkeit zu gewährleisten, empfiehlt es sich, mehrere StoreFront-Server zu implementieren.
- Konfigurieren Sie den externen Lastenausgleich für Failover zwischen den Servern, um sicherzustellen, dass Benutzer ständigen Zugriff auf Anwendungen und Desktops haben.



Remote-PC-Zugriff

Sep 29, 2015

Mit Remote-PC-Zugriff kann ein Endbenutzer sich remote von jedem Standort aus an einem physischen Windows-PC im Büro anmelden. Der Virtual Delivery Agent (VDA) wird auf dem Büro-PC installiert. Er registriert sich bei dem Delivery Controller und verwaltet die HDX-Verbindung zwischen dem PC und den Clientgeräten des Endbenutzers. Remote-PC-Zugriff unterstützt ein Self-Service Modell. Nach der Einrichtung der Positivliste der Maschinen, die für Benutzer zugänglich sind, können die entsprechenden Benutzer ihre Büro-PCs einer Site ohne Eingriff des Administrators hinzufügen. Citrix Receiver wird auf ihrem Clientgerät ausgeführt und ermöglicht Zugriff auf alle Anwendungen und Daten auf dem Büro-PC von der Remote-PC-Zugriff-Desktopsitzung aus.

Ein Benutzer kann mehrere Desktops haben, einschließlich mehrere physische PCs oder eine Kombination von physischen PCs und virtuellen Desktops.

Hinweis: Remote-PC-Zugriff gilt nur für XenDesktop-Lizenzen. Sitzungen verbrauchen Lizenzen genauso wie andere XenDesktop-Sitzungen.

Überlegungen zu Active Directory:

- Bevor Sie die Remote-PC-Bereitstellungssite konfigurieren, richten Sie die Organisationseinheiten und Sicherheitsgruppen ein und erstellen Sie dann Benutzerkonten. Verwenden Sie diese Konten, um die Benutzer für die Bereitstellungsgruppen anzugeben, die für Remote-PC-Zugriff verwendet werden.
- Wenn Sie Active Directory modifizieren, nachdem eine Maschine dem Maschinenkatalog hinzugefügt wurde, wird diese Zuweisung von Remote-PC-Zugriff nicht erneut ausgewertet. Sie können, falls erforderlich, Maschinen manuell einem anderen Katalog zuweisen.
- Wenn Sie Organisationseinheiten verschieben oder löschen, sind die für Remote-PC-Zugriff verwendeten möglicherweise nicht mehr aktuell. VDAs sind u. U. (wenn überhaupt) nicht mehr dem am besten geeigneten Maschinenkatalog bzw. der am besten geeigneten Bereitstellungsgruppe zugewiesen.

Überlegungen zu Maschinenkatalogen und Bereitstellungsgruppen:

- Eine Maschine kann nur jeweils einem Maschinenkatalog und einer Bereitstellungsgruppe zugewiesen sein.
- Sie können Maschinen in einen oder mehrere Remote PC-Zugriff-Maschinenkataloge einfügen.
- Wenn Sie Maschinenkonten für den Maschinenkatalog auswählen, wählen Sie die niedrigste zutreffende Organisationseinheit, um einen möglichen Konflikt mit Maschinen eines anderen Maschinenkatalogs zu vermeiden. Beispiel: Im Fall von Bank/Bankbeamte/Kassierer, wählen Sie Kassierer.
- Sie können alle Maschinen eines Remote PC-Maschinenkatalogs über eine oder mehrere Bereitstellungsgruppen zuweisen. Wenn eine Benutzergruppe beispielsweise bestimmte Richtlinieninstellungen und eine andere Gruppe andere Einstellungen benötigt, können Sie die Benutzer unterschiedlichen Bereitstellungsgruppen zuweisen und so die HDX-Richtlinien nach Bereitstellungsgruppe filtern.
- Wenn die Benutzerbetreuung in Ihrer IT-Infrastruktur auf dem geografischen Standort, der Abteilung oder einer anderen Kategorie beruht, können Sie Maschinen und Benutzer entsprechend gruppieren und so die delegierte Administration ermöglichen. Stellen Sie sicher, dass jeder Administrator über Berechtigungen für die relevanten Maschinenkataloge und die entsprechenden Bereitstellungsgruppen hat.
- Erstellen Sie für Benutzer mit Büro-PCs mit Windows XP einen separaten Maschinenkatalog und eine separate Bereitstellungsgruppe für diese Systeme. Aktivieren Sie bei der Auswahl von Maschinenkonten für diesen Katalog in Studio das Kontrollkästchen für Maschinen, die unter Windows XP ausgeführt werden.

Überlegungen zur Bereitstellung:

- Sie können eine Remote-PC-Zugriff-Bereitstellung erstellen und dann später konventionelle Virtual Desktop

Infrastructure-Desktops (VDI) oder Anwendungen hinzufügen. Sie können auch Remote-PC-Zugriff-Desktops einer vorhandenen VDI-Bereitstellung hinzufügen.

- Erwägen Sie eine Aktivierung der Windows-Remoteunterstützung bei der Installation des VDA auf Büro-PCs. Diese Option ermöglicht es den Helpdeskteams, die Director verwenden, Benutzersitzungen mit der Windows-Remoteunterstützung anzuzeigen und zu steuern.
- Legen Sie fest, wie Sie den VDA auf jedem Büro-PC bereitstellen möchten. Citrix empfiehlt, eine elektronische Softwareverteilung zu verwenden, z. B. Active Directory-Skripts und Microsoft System Center Configuration Manager. Das Installationsmedium enthält Active Directory-Beispielskripts.
- Jeder Büro-PC muss in der Domäne per Netzkabel verbunden sein.
- Windows 7 Aero wird für Büro-PCs unterstützt, ist aber nicht erforderlich.
- Tastatur und Maus müssen direkt am PC oder Laptop angeschlossen werden, nicht am Bildschirm oder an anderen Komponenten, die ausgeschaltet werden können. (Wenn Sie Eingabegeräte an solche Komponenten (z. B. Bildschirme) anschließen müssen, sollten diese nicht ausgeschaltet werden.)
- Informationen zum Remotezugriff mit Smartcards finden Sie unter [Systemanforderungen für Smartcards](#).
- Remote-PC-Zugriff kann auf den meisten Laptop-Computern verwendet werden. Für einen besseren Zugriff und die optimale Verbindung konfigurieren Sie die Energiesparoptionen bei Laptops wie die eines Desktop-PCs. Beispiel:
 - Deaktivieren Sie den Ruhezustand.
 - Deaktivieren Sie den Standbymodus.
 - Legen Sie die Aktion beim Schließen des Deckels auf Keine Aktion fest.
 - Stellen Sie die Aktion Beim Drücken des Netzschalters auf Herunterfahren.
 - Deaktivieren Sie die Energiesparfunktionen der Grafikkarte.
 - Deaktivieren Sie die Energiesparfunktionen der Netzwerkkarte.
 - Deaktivieren Sie Akku-Energiespartechnologien.

Folgendes wird für Remote-PC-Zugriff-Geräte nicht unterstützt:

- An-/Ausdocken des Laptops.
- KVM-Switches oder andere Komponenten, die eine Sitzung trennen.
- Hybrid-PCs (einschließlich All-in-One- und NVIDIA Optimus-Laptops und -PCs).
- Installieren Sie Citrix Receiver auf jedem Clientgerät mit Remotezugriff auf den Büro-PC.
- Mehrere Benutzer mit Remotezugriff auf denselben Büro-PC sehen in Receiver dasselbe Symbol. Wenn ein Benutzer eine Remoteanmeldung am PC vornimmt, wird diese Ressource anderen Benutzern als nicht verfügbar angezeigt.

Folgende XenDesktop-Features werden für Remote-PC-Zugriff-Bereitstellungen nicht unterstützt:

- Erstellen von Masterimages und virtuellen Maschinen
- Bereitstellen gehosteter Anwendungen
- Personal vDisks
- Clientordnerumleitung

Wake-On-LAN

Remote-PC-Zugriff unterstützt Wake-On-LAN, sodass physische PCs remote eingeschaltet werden können. Dieses Feature ermöglicht es Benutzern, ihre Büro-PCs ausgeschaltet zu lassen, wenn diese nicht verwendet werden, um Energiekosten zu sparen. Außerdem ist ein Remotezugriff möglich, wenn Maschinen unabsichtlich ausgeschaltet wurden, z. B. bei witterungsbedingten Problemen.

Das Wake-On-LAN-Feature von Remote-PC-Zugriff wird auf Folgendem unterstützt:

- PCs, die Intel Active Management Technology (AMT) unterstützen.
- PCs, auf denen Wake-On-LAN im BIOS aktiviert ist.

Sie müssen Microsoft System Center Configuration Manager (ConfigMgr) 2012 konfigurieren, um das Wake-On-LAN-Feature zu verwenden. ConfigMgr bietet Zugriff auf AMT-Energiebefehle für den Computer und Unterstützung für den Aktivierungsproxy sowie Magic Packets. Wenn Sie anschließend mit Studio eine Bereitstellung für den Remote-PC-Zugriff erstellen (bzw. wenn Sie eine andere Energieverwaltungsverbindung für den Remote-PC-Zugriff hinzufügen), aktivieren Sie die Energieverwaltung und geben Sie ConfigMgr-Zugriffsinformationen an.

Beachten Sie außerdem Folgendes:

- Die Verwendung der AMT-Energieverwaltung wird aus Gründen der Sicherheit und Sitzungszuverlässigkeit zwar bevorzugt, es werden jedoch auch zwei Methoden ohne AMT unterstützt: ConfigMgr-Aktivierungsproxy und Magic Packets.
- Nur AMT-fähige Maschinen: Das Wake-On-LAN-Feature unterstützt auch die Aktionen "Herunterfahren erzwingen" und "Neustart erzwingen" von Studio und Director. Außerdem ist in StoreFront und Receiver eine Neustartaktion verfügbar.

Weitere Informationen finden Sie unter [Microsoft System Center Configuration Manager und Wake-On-LAN-Feature für den Remote-PC-Zugriff](#) und [Bereitstellen von Remote-PC-Zugriff für Benutzer](#).

Erstellen einer neuen Umgebung

Sep 29, 2015

Richten Sie eine Umgebung in einer öffentlichen oder privaten Cloud ein oder verwenden Sie Virtualisierungsressourcen (z. B. XenServer, Hyper-V oder VMware).

Installieren

Sie können die Komponenten über eine assistentengestützte grafische Oberfläche installieren oder über eine Befehlszeilenoberfläche, die eine skriptbasierte Installation ermöglicht. Bei beiden Methoden werden die meisten Voraussetzungen automatisch installiert.

Wichtig: Vor Beginn einer Installation sollten Sie die Checkliste zum [Vorbereiten der Installation](#) durchgehen.

1. Installieren der Kernkomponenten: Delivery Controller, Citrix Studio, Citrix Director, Citrix Lizenzserver und Citrix StoreFront.
2. Erstellen Sie eine Site in Studio.
3. Wenn Sie Provisioningtools (z. B. Maschinenerstellungsdienste oder Provisioning Services) zum Erstellen von VMs von einem Masterimage verwenden, installieren Sie einen VDA für Desktopbetriebssysteme. Andernfalls installieren Sie VDA für Desktopbetriebssysteme oder für Serverbetriebssysteme direkt auf jeder Maschine.

Installieren Sie für Remote-PC-Zugriff-Bereitstellungen einen VDA für Desktopbetriebssysteme auf dem Büro-PC. Citrix empfiehlt, dass Sie vorhandene ESD-Methoden (Electronic Software Distribution) und die Befehlszeilenschnittstelle des VDA-Installationsprogramms verwenden.

4. Folgen Sie nach der Installation der Komponenten und dem Erstellen einer Site den Anleitungen in Studio zum Erstellen von Maschinenkatalogen und Bereitstellungsgruppen.

Einrichten einer Cloudumgebung

Dieses Release unterstützt das Provisioning von Serverbetriebssystem- und [Server-VDI](#)-Maschinen auf Cloudplattformen. Mit dieser Softwareplattform lassen sich Rechenressourcen poolen, um öffentliche oder private Infrastructure-as-a-Service (IaaS)-Clouds sowie Hybrid-IaaS-Clouds zu erstellen.

Sie können eine der folgenden Cloudbereitstellungslösungen auswählen:

- Amazon Web Services (AWS): Informationen finden Sie unter [Bereitstellen von XenApp 7.5 und XenDesktop 7.5 mit Amazon VPC](#).
- Citrix CloudPlatform: Informationen finden Sie unter [XenApp und XenDesktop – Konzepte und Bereitstellung auf CloudPlatform](#).

Verwalten von virtuellen Maschinen mit Microsoft System Center Virtual Machine Manager

Konfigurieren Sie das System wie in diesem Abschnitt beschrieben, wenn Sie Hyper-V mit Microsoft System Center Virtual Machine Manager (VMM) zur Bereitstellung von virtuellen Maschinen in der Umgebung verwenden.

Systemanforderungen

Bevor Sie virtuelle Maschinen (VMs) erstellen, stellen Sie sicher, dass die Umgebung die unter [Systemanforderungen für XenApp 7.5 und XenDesktop 7.5](#) aufgeführten Anforderungen erfüllt.

Unterstützung für virtuelle Maschinen

Beachten Sie Folgendes:

- Dieses Release unterstützt nur virtuelle Maschinen der ersten Generation mit VMM 2012 R2. Virtuelle Maschinen der zweiten Generation werden nicht unterstützt.
- Virtuelle Maschinen der zweiten Generation werden weder für Bereitstellungen mit Maschinenerstellungsdiensten (MCS) noch mit Provisioning Services unterstützt. Beim Erstellen von VMs in Studio mit Maschinenerstellungsdiensten oder Provisioning Services werden VMs der zweiten Generation nicht in der Auswahlliste für eine Master-VM aufgeführt.
- Auf VMs der zweiten Generation ist der sichere Start standardmäßig aktiviert und verhindert, dass der VDA richtig funktioniert.

Installieren und Konfigurieren eines Hypervisors

1. Installieren Sie Microsoft Hyper-V Server und VMM auf Ihren Servern. Alle Delivery Controller in Ihrer Umgebung müssen in derselben Gesamtstruktur sein wie die VMM-Server.
2. Installieren Sie die System Center Virtual Machine Manager-Konsole auf allen Delivery Controllern der Maschinenumgebung.
3. Überprüfen Sie die folgenden Kontoinformationen:
 - Das Konto, das Sie zum Erstellen von Hosts in Studio verwenden, ist ein VMM-Administrator oder delegierter VMM-Administrator für die relevanten Hyper-V-Maschinen. Wenn dieses Konto nur über die delegierte Administratorrolle in VMM verfügt, werden die Speicherdaten in Studio beim Erstellen des Hosts nicht aufgeführt.
 - Das Benutzerkonto, das für die Studio-Integration verwendet wird, muss auch Mitglied der lokalen Administratorsicherheitsgruppe auf jedem Hyper-V-Server sein, um zur Lebenszyklusverwaltung von VMs (z. B. VMs erstellen, aktualisieren und löschen) berechtigt zu sein. Hinweis: Die Installation des Delivery Controller direkt auf einem Server, auf dem Hyper-V ausgeführt wird, wird nicht unterstützt.

Erstellen einer Master-VM

Über eine Master-VM werden Benutzerdesktops bereitgestellt.

1. Installieren Sie den Virtual Desktop Agent auf der Master-VM und wählen Sie dabei die Option zur Desktopoptimierung aus. Hierdurch wird die Leistung der Benutzerdesktops verbessert, indem verschiedene Windows-Funktionen neu konfiguriert werden, die mit virtuellen Desktops nicht kompatibel sind oder nicht benötigt werden.
2. Erstellen Sie einen Snapshot der Master-VM, um diesen als Sicherungskopie zu verwenden. Weitere Informationen finden Sie unter [Vorbereiten eines Masterimages](#).

Erstellen virtueller Desktops

Wenn Sie die VMs mit Citrix Studio statt über einen vorhandenen Maschinenkatalog erstellen, führen Sie in Studio den Bereitstellungsassistenten aus und erstellen Sie die virtuellen Desktops:

1. Wählen Sie auf der Seite Host zuerst Virtuelle Maschinen aus und legen Sie dann Microsoft virtualization als Hosttyp fest.
2. Geben Sie die Dienstadresse als vollständig qualifizierten Domännennamen des Hostservers ein.
3. Geben Sie die Anmeldeinformationen für das zuvor erstellte Administratorkonto ein, das Berechtigungen zum Erstellen von neuen VMs enthält.
4. Wählen Sie im Dialogfeld Hostdetails den Cluster oder eigenständigen Host aus, der beim Erstellen der neuen VMs verwendet werden soll. Wichtig: Sie müssen auch dann zu einem Cluster oder eigenständigen Host navigieren und diesen auswählen, wenn Sie eine Bereitstellung mit einem einzelnen Hyper-V-Host verwenden.

Upgrade von einer früheren Version von VMM

Upgrade von VMM 2012

Beachten Sie die folgenden Kombinationen der Betriebssystemversionen von Komponenten, wenn Sie ein Upgrade von VMM 2012, VMM 2012 SP1 und VMM 2012 R2 durchführen:

- Informationen zu den Anforderungen von VMM- und Hyper-V-Hosts finden Sie unter <http://technet.microsoft.com/en-us/library/gg610649.aspx>.
- Informationen zu den Anforderungen für die VMM-Konsole finden Sie unter <http://technet.microsoft.com/en-us/library/gg610640.aspx>. Hinweis: Ein gemischter Hyper-V-Cluster wird nicht unterstützt. Ein gemischter Cluster ist beispielsweise einer, bei dem eine Hälfte des Clusters Hyper-V 2008 und die andere Hälfte Hyper-V 2012 ausführen.

Upgrade von VMM 2012 SP1 auf VMM 2012 R2

Bei XenDesktop 7 unter VMM 2012 SP1 ist es wichtig, die Schritte in der richtigen Reihenfolge durchzuführen, damit XenDesktop weiterhin ohne Ausfallzeit ausgeführt werden kann.

Empfohlene Reihenfolge für das Upgrade von Komponenten:

1. Upgrade von XenDesktop auf 7.5 oder von XenApp auf 7.5 (jetzt XenDesktop 7.5 oder XenApp 7.5 und VMM 2012 SP1)
2. Upgrade von VMM 2012 SP1 auf 2012 R2 (jetzt XenDesktop 7.5 oder XenApp 7.5 und VMM 2012 R2)

Upgrade von VMM 2008 R2 auf VMM 2012 SP1

Führen Sie beim Upgrade von XenDesktop 5.6 unter VMM 2008 R2 die Schritte in dieser Reihenfolge durch, damit XenDesktop weiterhin ohne Ausfallzeit ausgeführt werden kann.

Empfohlene Reihenfolge für das Upgrade von Komponenten:

1. Upgrade von VMM auf 2012 (jetzt XenDesktop 5.6 und VMM 2012)
2. Upgrade von XenDesktop auf 7.x (jetzt XenDesktop 7.x und VMM 2012)
3. Upgrade von VMM von 2012 auf 2012 SP1 (jetzt XenDesktop 7.x und VMM 2012 SP1)

Unterstützung für Microsoft System Center Virtual Machine Manager

Dieses Release unterstützt Folgendes:

- **VMM 2012:** Bietet bessere Verwaltungsfunktionen und ermöglicht die Verwaltung des gesamten virtualisierten Datacenters sowie der virtuellen Maschinen. Dieses Release organisiert das Patchen von Clusterhosts sowie die Integration mit Windows Server Update Services und ermöglicht somit das Erstellen eines Basisplans für Patches, die jeder Host benötigt.
- **VMM 2012 SP1:** Bietet Leistungsverbesserungen für Maschinenerstellungsdienste (MCS), wenn Sie SMB 3.0 auf Dateiservern mit freigegebenen Clustervolumen und Storage Area Networks (SANs) verwenden. Diese Dateifreigaben bieten kostengünstiges Zwischenspeichern und reduzierte E/A im SAN-Speicher mit verbesserter Leistung.
- **VMM 2012 R2:** Ermöglicht die angemessene Verwaltung von wichtigen Windows Server 2012 R2-Funktionen, wie das Ausführen von VM-Snapshots, dynamische VHDX-Größenanpassung und Speicherplätze.

Maschinenerstellungsdienste (MCS) auf SMB 3-Dateifreigaben

Bei Maschinenkatalogen, die über MSC auf SMB 3-Dateifreigaben für VM-Speicher erstellt wurden, müssen Sie darauf achten, dass die Anmeldeinformationen wie folgt eingerichtet werden, damit bei Aufrufen von der Hypervisor Communications Library (HCL) des Delivery Controllers die Verbindung mit dem SMB-Speicher hergestellt werden kann:

- Die VMM-Benutzeranmeldeinformationen müssen vollständigen Lese-/Schreibzugriff auf den SMB-Speicher umfassen.
- Speichervorgänge auf dem virtuellen Datenträger werden bei Vorgängen im Lebenszyklus der VM über den Hyper-V-Server mit den VMM-Anmeldeinformationen durchgeführt.

Hinweis: Wenn Sie den SMB-Speicher verwenden, aktivieren Sie "CredSSP" über den Delivery Controller für die einzelnen Hyper-V-Maschinen, wenn Sie mit VMM 2012 SP1 mit Hyper-V unter Windows Server 2012 arbeiten. Weitere Informationen finden Sie unter

— Aktivieren von CredSSP

im KB-Artikel <http://support.citrix.com/article/CTX137465>.

Über eine standardmäßige PowerShell V3-Remotesitzung kann HCL die Verbindung mit der Hyper-V-Maschine mittels der Funktion "CredSSP" herstellen. Diese Funktion übergibt die Anmeldeinformationen der Benutzer an die Hyper-V-Maschine (mit Kerberos-Verschlüsselung). Die PowerShell-Befehle in dieser Sitzung auf der Remotemaschine mit Hyper-V verwenden die angegebenen Anmeldeinformationen (in diesem Fall, die der VMM-Benutzer), sodass eine ordnungsgemäße Kommunikation mit dem Speicher gewährleistet wird.

Die folgenden Aufgaben verwenden PowerShell-Skripts der Delivery Controller-HCL, die an die Hyper-V-Maschine zur Verwendung mit SMB 3.0-Speicher gesendet werden.

Konsolidieren des Masterimages

Vom Masterimage wird ein neues MCS-Provisioningschema erstellt. Die Master-VM wird durch dieses Schema geklont und vereinfacht, damit sie zum Erstellen neuer VMs aus dem neu erstellten Datenträger bereit ist (die Abhängigkeit zur ursprünglichen Master-VM wird entfernt).

Beispiel

```
$ims = Get-WmiObject -class $class -namespace "root\virtualization\v2"; $result = $ims.ConvertVirtualHardDisk($diskName, $vhdaText) $result
```

Erstellen eines differenzierenden Datenträgers

Erstellt einen differenzierenden Datenträger aus dem Masterimage, das durch Konsolidierung des Masterimages generiert wurde. Der differenzierende Datenträger wird dann an eine neue VM angeschlossen.

Beispiel

```
$ims = Get-WmiObject -class $class -namespace "root\virtualization\v2"; $result = $ims.CreateVirtualHardDisk($vhdaText); $result
```

Upload von Identitätsdisks

Die Hypervisor Communications Library (HCL) kann die Identitätsdisk nicht direkt in den SMB-Speicher hochladen. Daher muss die Identitätsdisk von der Hyper-V-Maschine hochgeladen und in den Speicher kopiert werden. Da die Hyper-V-Maschine die Disk nicht auf dem Delivery Controller lesen kann, muss sie von der HCL zuerst wie folgt über die Hyper-V-Maschine kopiert werden:

1. Upload der Identitätsdisk durch HCL auf die Hyper-V-Maschine über die Administratorfreigabe.
2. Die Disk wird von der Hyper-V-Maschine über ein PowerShell-Skript, das in der Remote-PowerShell V3-Sitzung ausgeführt wird, in den SMB-Speicher kopiert. Auf der Hyper-V-Maschine wird ein Ordner erstellt, dessen Berechtigungen nur für den VMM-Benutzer gesperrt sind (über die remote PowerShell-Verbindung).
3. Die HCL löscht die Datei aus der Administratorfreigabe.
4. Die HCL führt den Upload der Identitätsdisk auf die Hyper-V-Maschine durch. Durch die Remote-PowerShell-Sitzung wird die Identitätsdisk in den SMB-Speicher kopiert und aus der Hyper-V-Maschine gelöscht.

Hinweis: Falls der Ordner der Identitätsdisk gelöscht wird, wird er neu erstellt, damit er zur Wiederverwendung verfügbar ist.

Download von Identitätsdisks

Wie beim Upload wird die Identitätsdisk über die Hyper-V-Maschine an die HCL übergeben. Beim folgenden Prozess wird, falls noch nicht vorhanden, ein Ordner erstellt, der nur VMM-Benutzerberechtigungen auf dem Hyper-V-Server hat.

1. Die Disk wird von der Hyper-V-Maschine aus dem SMB-Speicher in den lokalen Hyper-V-Speicher kopiert, und zwar über ein PowerShell-Skript, das in der Remote-PowerShell V3-Sitzung ausgeführt wird.
2. Die HCL liest die Disk aus der Administratorfreigabe der Hyper-V-Maschine in den Speicher.
3. Die HCL löscht die Datei aus der Administratorfreigabe.

Erstellen einer persönlichen vDisk

Wenn der Administrator die VM in einem Personal vDisk-Maschinenkatalog erstellt, muss eine leere Disk (persönliche vDisk) erstellt werden.

Der Aufruf zum Erstellen einer leeren Disk erfordert keinen direkten Zugriff auf den Speicher. Wenn Sie PvDs auf anderen Speichern als dem Haupt- oder dem Betriebssystemdatenträger haben, verwenden Sie Remote-PowerShell zum Erstellen der PvD in einem Ordner mit dem gleichen Namen wie die VM, aus dem sie erstellt wurde. Verwenden Sie Remote-PowerShell nicht mit CSV oder LocalStorage. VMM-Befehlsfehler werden vermieden, wenn zuerst das Verzeichnis und dann die leere Disk erstellt werden.

Führen Sie auf der Hyper-V-Maschine `mkdir` an dem Speicher aus.

Weitere Informationen zum Verwenden des SDKs finden Sie unter [Informationen zum XenApp- und XenDesktop-SDK](#).

Integration von Microsoft System Center Configuration Manager

Bei Sites, in denen der Zugriff auf Anwendungen und Desktops auf physischen Geräten mit System Center Configuration Manager (Configuration Manager) 2012 verwaltet wird, kann diese Verwendung über die Integrationsoptionen auf XenApp bzw. XenDesktop ausgeweitet werden.

- **Citrix Connector 7.5 für Configuration Manager 2012:** Citrix Connector bildet eine Brücke zwischen Configuration Manager und XenApp bzw. XenDesktop. Mit Citrix Connector können Sie alltägliche Vorgänge in den mit Configuration Manager verwalteten physischen Umgebungen und den mit XenApp bzw. XenDesktop verwalteten virtuellen Umgebungen übergreifend vereinheitlichen. Weitere Informationen zum Connector finden Sie unter [Citrix Connector 7.5 für System Center Configuration Manager 2012](#).
- **Configuration Manager Wake Proxy-Feature:** Unabhängig davon, ob in der Umgebung Citrix Connector verwendet wird, ist für die Nutzung des Wake-On-LAN-Features für Remote-PC-Zugriff Configuration Manager erforderlich. Weitere Informationen finden Sie unter [Microsoft System Center Configuration Manager und Remote-PC-Zugriff-Wake-On-LAN](#).
- **XenApp- und XenDesktop-Eigenschaften:** XenApp- und XenDesktop-Eigenschaften ermöglichen die Identifizierung virtueller Citrix Desktops für die Verwaltung durch Configuration Manager. Diese Eigenschaften werden automatisch von Citrix Connector verwendet, können aber auch manuell konfiguriert werden. Die Eigenschaften werden im verbleibenden Teil dieses Abschnitts beschrieben.

Die Eigenschaften stehen in der Klasse **Citrix_virtualDesktopInfo** im Namespace `Root\Citrix\DesktopInformation` zur Verfügung.

Die folgenden Eigenschaften sind verfügbar. Die Namen der Eigenschaften stammen vom Anbieter für Windows-Verwaltungsinstrumentation (WMI):

- **AssignmentType:** legt den Wert **IsAssigned** fest. Gültige Werte für **AssignmentType** sind:
 - **ClientIP**

- **ClientName**
- **Keine**
- **User**: Dieser Wert legt **IsAssigned** auf **True** fest.
- **BrokerSiteName**: Site; gibt den gleichen Wert wie **HostIdentifier** zurück.
- **DesktopCatalogName**: dem Desktop zugeordneter Maschinenkatalog.
- **DesktopGroupName**: dem Desktop zugeordnete Bereitstellungsgruppe.
- **HostIdentifier**: Site; gibt den gleichen Wert wie **BrokerSiteName** zurück.
- **IsAssigned**: True = ordnet den Desktop einem Benutzer zu; False = zufälliger Desktop.
- **IsMasterImage**: ermöglicht Entscheidungen bezüglich der Umgebung. Beispielsweise können Sie Anwendungen auf dem Masterimage und nicht auf den bereitgestellten Maschinen installieren, besonders dann, wenn diese Maschinen auf Bootmaschinen in einem fehlerfreien Zustand sind. Zulässige Werte sind:
 - **True** auf einer virtuellen Maschine (VM), die als Masterimage verwendet wird (dieser Wert wird während der Installation festgelegt, basierend auf einer Auswahl während der Installation).
 - **Cleared** auf einer VM, die von diesem Image bereitgestellt wird.
- **IsVirtualMachine**: True für eine virtuelle Maschine, False für einen physischen Computer.
- **OSChangesPersist**: False, wenn das Betriebssystemimage des Desktops bei jedem Neustart in einen fehlerfreien Zustand versetzt wird, andernfalls True.
- **PersistentDataLocation**: der Speicherort, an dem Configuration Manager persistente Daten speichert. Benutzer haben hierauf keinen Zugriff.
- **PersonalvDiskDriveLetter**: Bei einem Desktop mit einer persönlichen vDisk ist dies der Laufwerksbuchstabe, den Sie der persönlichen vDisk zuweisen.

Die Eigenschaften **BrokerSiteName**, **DesktopCatalogName**, **DesktopGroupName** und **HostIdentifier** werden festgelegt, wenn der Desktop sich beim Controller registriert. Sie sind also bei Desktops, die noch nicht vollständig registriert sind, NULL.

Zum Sammeln der Eigenschaften führen Sie eine Hardwareinventur in Configuration Manager durch. Zum Anzeigen der Eigenschaften verwenden Sie den Ressourcen-Explorer von Configuration Manager. In diesen Fällen können die Namen Leerzeichen enthalten oder vom Eigenschaftsnamen etwas abweichen. **BrokerSiteName** kann z. B. als "Broker Site Name" angezeigt werden. Informationen zu den folgenden Aufgaben finden Sie unter [Citrix WMI Properties and System Center Configuration Manager 2012](#):

- Konfigurieren von Configuration Manager zum Sammeln von Citrix WMI-Eigenschaften vom Citrix VDA
- Erstellen abfragebasierter Gerätesammlungen mit Citrix WMI-Eigenschaften
- Erstellen globaler Bedingungen basierend auf Citrix WMI-Eigenschaften
- Verwenden globaler Bedingungen zum Definieren von Anforderungen für Anwendungsbereitstellungstypen

Sie können in der Microsoft-Klasse **CCM_DesktopMachine** im Namespace Root\ccm_vdi auch Microsoft-Eigenschaften verwenden. Weitere Informationen zu diesen Eigenschaften finden Sie in der Microsoft-Dokumentation.

Hinweis: Boolesche Eigenschaften in Configuration Manager 2012 können als 1 oder 0 statt True oder False angezeigt werden.
Microsoft System Center Configuration Manager und Remote-PC-Zugriff-Wake-On-LAN

Eine Beschreibung des Wake-On-LAN-Features von Remote-PC-Zugriff finden Sie unter [Remote-PC-Zugriff](#).

Zum Konfigurieren des Features führen Sie die folgenden Schritte aus, bevor Sie einen VDA auf den Büro-PCs installieren und mit Studio die Remote-PC-Zugriff-Bereitstellung erstellen oder aktualisieren:

- Konfigurieren Sie Microsoft System Center Configuration Manager (ConfigMgr) 2012 im Unternehmen und stellen Sie dann den ConfigMgr-Client für alle Remote-PC-Zugriff-Maschinen bereit. Planen Sie genug Zeit ein, damit der geplante SCCM-Bestandszyklus ausgeführt werden kann (oder erzwingen Sie ihn bei Bedarf manuell). Die Zugriffsanmeldeinformationen, die Sie in Studio zum Konfigurieren der Verbindung mit ConfigMgr festlegen, müssen Sammlungen im Geltungsbereich und die Rolle "Remotetoolsverantwortlicher" umfassen.

- Für die Unterstützung von Intel Active Management Technology (AMT) gilt Folgendes:
 - Die unterstützte Mindestversion auf dem PC ist AMT 3.2.1.
 - Stellen Sie den PC für die Verwendung von AMT mit Zertifikaten und zugehörigen Provisioningvorgängen bereit.
- Für die Unterstützung von ConfigMgr Wake Proxy bzw. Magic Packet gilt Folgendes:
 - Konfigurieren Sie Wake-On-LAN in den BIOS-Einstellungen aller PCs.
 - Zur Unterstützung von ConfigMgr Wake Proxy aktivieren Sie die entsprechende Option in ConfigMgr. Stellen Sie für jedes Subnetz des Unternehmens mit PCs, auf denen das Wake-On-LAN-Feature von Remote-PC-Zugriff verwendet wird, sicher, dass mindestens drei Maschinen als Sentinelmaschinen fungieren können.
 - Zur Unterstützung von Magic Packet konfigurieren Sie Netzwerkrouter und Firewalls so, dass Magic Packets vom Delivery Controller entweder per subnetzgesteuertem Broadcast oder Unicast gesendet werden können.

Nach der Installation des VDAs auf Büro-PCs aktivieren oder deaktivieren Sie die Energieverwaltung beim Erstellen der Remote-PC-Zugriff-Bereitstellung in Studio.

- Wenn Sie die Energieverwaltung aktivieren, geben Sie Verbindungsdetails an, d. h. ConfigMgr-Adresse und Anmeldeinformationen sowie einen Namen.
- Wenn Sie die Energieverwaltung nicht aktivieren, können Sie eine Energieverwaltungsverbindung (ConfigMgr-Verbindung) später hinzufügen und dann einen Remote-PC-Zugriff-Maschinenkatalog zur Aktivierung der Energieverwaltung bearbeiten und die neue Energieverwaltungsverbindung angeben.

Sie können eine Energieverwaltungsverbindung bearbeiten, um die Verwendung von ConfigMgr Wake Proxy und Magic Packets zu konfigurieren und die Paketübertragungsmethode zu ändern. Weitere Informationen finden Sie unter [Bereitstellen von Remote-PC-Zugriff für Benutzer](#).

Verwalten von virtuellen Maschinen mit VMware

Konfigurieren Sie das System wie in diesem Abschnitt beschrieben, wenn Sie virtuelle Maschinen mit VMware in der Umgebung bereitstellen.

Informationen zu unterstützten VMware-Versionen finden Sie unter [Systemanforderungen für XenApp 7.5 und XenDesktop 7.5](#).

Installieren und Konfigurieren des Hypervisors

1. Installieren Sie vCenter Server und die erforderlichen Verwaltungstools.
Hinweis: Der "Linked Mode"-Betrieb von vSphere vCenter wird nicht unterstützt.
2. Erstellen Sie ein VMware-Benutzerkonto mit den folgenden Mindestberechtigungen auf DataCenter-Ebene:
Hinweis: Dieses Konto verfügt über Berechtigungen zum Erstellen neuer VMs und wird verwendet, um mit vCenter zu kommunizieren.

SDK	Benutzeroberfläche
Datastore.AllocateSpace	Datastore > Allocate space
Datastore.Browse	Datastore > Browse datastore
Datastore.FileManagement	Datastore > Low level file operations
Network.Assign	Network > Assign network
Resource.AssignVMToPool	Resource > Assign virtual machine to resource pool
SystemAnonymous	Automatisch hinzugefügt.
SystemRead	Automatisch hinzugefügt.

System.View SDK	Automatisch hinzugefügt. Benutzeroberfläche
Task.Create	Tasks > Create task
VirtualMachine.Config.AddRemoveDevice	Virtual machine > Configuration > Add or remove device
VirtualMachine.Config.AddExistingDisk	Virtual machine > Configuration > Add existing disk
VirtualMachine.Config.AddNewDisk	Virtual machine > Configuration > Add new disk
VirtualMachine.Config.AdvancedConfig	Virtual machine > Configuration > Advanced
VirtualMachine.Config.CPUCount	Virtual machine > Configuration > Change CPU Count
VirtualMachine.Config.Memory	Virtual machine > Configuration > Memory
VirtualMachine.Config.RemoveDisk	Virtual machine > Configuration > Remove disk
VirtualMachine.Config.Resource	Virtual machine > Configuration > Change resource
VirtualMachine.Config.Settings	Virtual machine > Configuration > Settings
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	Virtual machine > Inventory > Create new
VirtualMachine.Inventory.CreateFromExisting	Virtual machine > Inventory > Create from existing
VirtualMachine.Inventory.Delete	Virtual machine > Inventory > Remove
VirtualMachine.Inventory.Register	Virtual machine > Inventory > Register
VirtualMachine.Provisioning.Clone	Virtual machine > Provisioning > Clone template
VirtualMachine.Provisioning.DiskRandomAccess	Virtual machine > Provisioning > Allow disk access
VirtualMachine.Provisioning.GetVmFiles	Virtual machine > Provisioning > Allow virtual machine download

VirtualMachine.Provisioning.PutVmFiles	Virtual machine > Provisioning > Allow virtual machine files upload
VirtualMachine.Provisioning.DeployTemplate	Virtual machine > Provisioning > Deploy template
VirtualMachine.Provisioning.MarkAsVM	Virtual machine > Provisioning > Mark as virtual machine
VirtualMachine.State.CreateSnapshot	- vSphere 5.0, Update 2 und vSphere 5.1, Update 1: Virtual machine > State > Create snapshot - vSphere 5.5: Virtual machine > Snapshot management > Create snapshot
VirtualMachine.State.RemoveSnapshot	- vSphere 5.0, Update 2 und vSphere 5.1, Update 1: Virtual machine > State > Remove snapshot - vSphere 5.5: Virtual machine > Snapshot management > Remove snapshot
VirtualMachine.State.RevertToSnapshot	- vSphere 5.0, Update 2 und vSphere 5.1, Update 1: Virtual machine > State > Revert to snapshot - vSphere 5.5: Virtual machine > Snapshot management > Revert to snapshot

3. Wenn Sie die erstellten VMs mit Tags kennzeichnen wollen, fügen Sie die folgenden Berechtigungen dem Benutzerkonto hinzu:

SDK	Benutzeroberfläche
Global.ManageCustomFields	Global > Manage custom attributes
Global.SetCustomField	Global > Set custom attribute

Um sicherzustellen, dass Sie ein sauberes Basisimage zum Erstellen neuer VMs verwenden, fügen Sie Tags den VMs hinzu, die mit den Maschinenerstellungsdiensten erstellt wurden. Damit schließen Sie sie aus der Liste der für Basisimages verfügbaren virtuellen Maschinen aus.

4. Um die vSphere-Kommunikation zu schützen, empfiehlt Citrix die Verwendung von HTTPS statt HTTP. HTTPS benötigt digitale Zertifikate. Citrix empfiehlt die Verwendung eines digitalen Zertifikats, das von einer Zertifizierungsstelle unter Berücksichtigung der Sicherheitsrichtlinie Ihrer Organisation erstellt wurde.

Wenn Sie kein digitales Zertifikat verwenden können, das von einer Zertifizierungsstelle ausgestellt wurde, können Sie das mit VMware installierte selbstsignierte Zertifikat verwenden, vorausgesetzt, die Sicherheitsrichtlinie Ihrer Organisation lässt dies zu. Das VMware vCenter-Zertifikat muss jedem Delivery Controller in der Umgebung hinzugefügt werden. Gehen Sie hierzu folgendermaßen vor:

- Fügen Sie den vollqualifizierten Domännennamen (FQDN) des Computers, auf dem vCenter Server ausgeführt wird, der Hostdatei auf dem Server im Verzeichnis %SystemRoot%/WINDOWS/system32/Drivers/etc/ hinzu. Dieser Schritt ist nur erforderlich, wenn der FQDN des Computers, auf dem vCenter Server ausgeführt wird, nicht bereits im Domännennamensystem vorhanden ist.
- Rufen Sie das vCenter-Zertifikat mit einer der folgenden Methoden ab:
 - Kopieren vCenter-Server:
 - Kopieren Sie die Datei rui.crt vom vCenter-Server zu einem Speicherort, auf den Ihre Delivery Controller zugreifen können. Der Standardspeicherort auf dem vCenter-Server ist c:\Documents and Settings\All Users\Applications Data\VMware\VMware VirtualCenter\SSL\

2. Navigieren Sie auf dem Delivery Controller auf den Speicherort des exportierten Zertifikats und öffnen Sie die Datei rui.crt.
- Laden Sie das Zertifikat über einen Webbrowser herunter. Bei Verwendung von Internet Explorer müssen Sie (abhängig von Ihrem Benutzerkonto) ggf. in Internet Explorer mit der rechten Maustaste klicken und "Als Administrator ausführen" wählen, um das Zertifikat herunterzuladen und zu installieren.
 1. Öffnen Sie einen Webbrowser und stellen Sie eine sichere Webverbindung zu dem vCenter-Server her, z. B.
https://server1.domain1.com
 2. Akzeptieren Sie die Sicherheitswarnungen.
 3. Klicken Sie auf die Adressleiste, in der der Zertifikatsfehler angezeigt wird.
 4. Zeigen Sie das Zertifikat an und klicken Sie auf die Registerkarte Details.
 5. Wählen Sie In Datei kopieren und exportieren Sie im CER-Format; geben Sie an entsprechender Stelle einen Namen an.
 6. Speichern Sie das exportierte Zertifikat.
 7. Navigieren Sie auf den Speicherort des exportierten Zertifikats und öffnen Sie die CER-Datei.
- Direkter Import von Internet Explorer, der als Administrator ausgeführt wird:
 1. Öffnen Sie einen Webbrowser und stellen Sie eine sichere Webverbindung zu dem vCenter-Server her, z. B.
https://server1.domain1.com
 2. Akzeptieren Sie die Sicherheitswarnungen.
 3. Klicken Sie auf die Adressleiste, in der der Zertifikatsfehler angezeigt wird.
 4. Zeigen Sie das Zertifikat an.
3. Importieren des Zertifikats auf jedem Delivery Controller in den Zertifikatspeicher:
 1. Klicken Sie auf Zertifikat installieren, wählen Sie Lokaler Computer aus und klicken Sie dann auf Weiter.
 2. Wählen Sie Alle Zertifikate in folgendem Speicher speichern aus und klicken Sie dann auf Durchsuchen.
Bei Verwendung von Windows Server 2008 R2:
 1. Aktivieren Sie das Kontrollkästchen Physischen Speicher anzeigen.
 2. Erweitern Sie Vertrauenswürdige Personen.
 3. Wählen Sie Lokaler Computer.
 4. Klicken Sie auf Weiter und dann auf Fertig stellen.
 Bei Verwendung von Windows Server 2012 oder Windows Server 2012 R2:
 1. Wählen Sie Vertrauenswürdige Personen und klicken Sie auf OK.
 2. Klicken Sie auf Weiter und dann auf Fertig stellen.

Wichtig: Wenn Sie den Namen des vSphere Servers nach der Installation ändern, müssen Sie ein neues selbstsigniertes Zertifikat auf diesem Server erstellen, bevor Sie das neue Zertifikat importieren.

Erstellen einer Master-VM

Über eine Master-VM werden Benutzern Desktops und Anwendungen bereitgestellt.

1. Installieren Sie VDA auf der Master-VM und wählen Sie die Option zur Desktopoptimierung aus. Hierdurch wird die Leistung der Desktops und Anwendungen der Benutzer verbessert, indem verschiedene Windows-Funktionen neu konfiguriert werden, die mit virtuellen Desktops nicht kompatibel sind oder nicht benötigt werden.
2. Erstellen Sie einen Snapshot der Master-VM, um diesen als Sicherungskopie zu verwenden. Weitere Informationen finden Sie unter [Vorbereiten eines Masterimages](#).

Erstellen virtueller Desktops

Wenn Sie VMs mit Studio erstellen statt einen vorhandenen Maschinenkatalog auszuwählen, stellen Sie sicher, dass Sie beim Einrichten der Hostinginfrastruktur für die Erstellung virtueller Desktops die folgenden Informationen eingeben:

1. Klicken Sie auf der Seite Hostverbindung auf VMware vSphere® als Hosttyp.
2. Geben Sie die Adresse des Zugriffspunkts für das vCenter SDK ein.
Beispiel: https://vmware.example.com/sdk.
3. Geben Sie die Anmeldeinformationen für das zuvor eingerichtete VMware-Konto ein, das Berechtigungen zum Erstellen neuer VMs hat.
Geben Sie den Benutzernamen für das Konto im Format Domäne/Benutzername ein.

Vorbereiten der Installation

Sep 29, 2015

In den folgenden Tabellen finden Sie Listen mit Aufgaben, die ausgeführt und Aktionen, die berücksichtigt werden müssen, bevor Sie die Kernkomponenten (Delivery Controller, Citrix Studio, Citrix Director, Citrix Lizenzserver und StoreFront) und Virtual Delivery Agents (VDAs) installieren.

Vorbereitungen der Kernkomponenten und allgemeine Installationsvorbereitungen

 Beschreibung
Zu Beginn: - Lesen Sie die Abschnitte im Bereich <i>— Planen</i>. - Informieren Sie sich über bekannte Probleme, die bei der Installation auftreten können. - Wenn Sie Komponenten in einer Cloudumgebung installieren, lesen Sie Einrichten der Cloudumgebung.
Legen Sie fest, wo die Komponenten installiert werden, und bereiten Sie die Maschinen und Betriebssysteme vor. - Lesen Sie die Systemanforderungen für XenApp 7.5 und XenDesktop 7.5 für die unterstützten Betriebssysteme und Versionen für Controller, Studio, Director, Virtualisierungsressourcen und VDAs. In den Abschnitten zu den Systemanforderungen für Citrix StoreFront und Citrix Lizenzserver finden Sie Angaben zu den unterstützten Plattformen. - Sie können die Kernkomponenten auf dem gleichen Server oder auf unterschiedlichen Servern installieren. Um beispielsweise eine kleinere Bereitstellung remote zu verwalten, können Sie Studio auf einer anderen Maschine installieren als den Server, auf dem der Controller installiert wurde. Um zukünftige Erweiterungen zu ermöglichen, sollten Sie die Komponenten auf verschiedenen Servern installieren, z. B. den Lizenzserver und Director auf unterschiedlichen Servern. - Sie können den Delivery Controller und den Virtual Delivery Agent für Windows-Serverbetriebssysteme auf demselben Server installieren. Starten Sie den Installer und wählen Sie den Delivery Controller (sowie alle weiteren Kernkomponenten für diese Maschine). Starten Sie dann den Installer erneut und wählen Sie den Virtual Delivery Agent für Windows-Serverbetriebssysteme. - Stellen Sie sicher, dass für jedes Betriebssystem die neuesten Updates ausgeführt wurden. - Stellen Sie sicher, dass bei allen Maschinen die Systemuhren synchronisiert sind. Die Synchronisierung wird für die Kerberos-Infrastruktur benötigt, die die Kommunikation zwischen den Maschinen sichert. - Komponenten werden standardmäßig unter C:\Programme\Citrix installiert. Sie können einen anderen Speicherort während der Installation angeben. Stellen Sie jedoch sicher, dass dieser Speicherort den Netzwerkdienst ausführen kann. - Die meisten Komponentenvoraussetzungen werden automatisch installiert. Ausnahmen hiervon enthält der Abschnitt <i>— Systemanforderungen</i>.
Legen Sie den Installationsort der SQL-Serversoftware für die Sitekonfigurationsdatenbank fest. - Standardmäßig wird SQL Server 2012 Express automatisch auf dem Server installiert, wenn Sie den Controller installieren, sofern keine andere Instanz erkannt wird. Sie können auch eine unterstützte Version von SQL Server auf dem Server oder auf einem anderen Server separat installieren. In diesem Fall muss die SQL Server-Software nicht vor der Installation der Kernkomponenten installiert werden, aber sie muss installiert werden, bevor Sie die Site erstellen. - Überprüfen Sie anhand der Abschnitte <i>— Planen</i>, was bei Datenbanken zu berücksichtigen ist, und richten Sie andere unterstützte Redundanzinfrastrukturen ein. Wichtig: Die Windows-Authentifizierung ist für Verbindungen zwischen dem Controller und der SQL Server-Datenbank erforderlich.
Legen Sie fest, wie Ports geöffnet werden. Standardmäßig werden die folgenden Ports automatisch geöffnet, wenn der Windows-Firewalldienst ausgeführt wird, selbst wenn die Firewall nicht aktiviert ist. Sie können diese Standardaktion deaktivieren und die Ports manuell öffnen, wenn Sie die Firewall eines Drittanbieters oder keine Firewall verwenden, oder wenn Sie die Ports einfach selbst öffnen möchten. - Controller: TCP 80, 443 - Director: TCP 80, 443 - Lizenzserver: TCP 7279, 8082, 8083, 27000 - StoreFront: TCP 80, 443 Tipp: Vollständige Portinformationen finden Sie unter CTX101810.
Konfigurieren Sie Ihre Active Directory-Domäne. - Auf den Maschinen, auf denen Sie die Kernkomponenten installieren, müssen Sie sowohl ein Domänenbenutzer als auch ein lokaler Administrator sein. - Versuchen Sie nicht, Komponenten auf einem Domänencontroller zu installieren. - Weitere Informationen finden Sie unter <i>— Planen</i> in den Abschnitten zu Active Directory. Weitere Informationen finden Sie in der Microsoft Dokumentation. Wenn Sie den Lizenzserver installieren, wird das Benutzerkonto automatisch zum Volladministrator auf dem Lizenzserver.
Bei der Installation von Director müssen Sie entscheiden, ob Sie die Spiegelungsfunktion von Director verwenden möchten, bei der Windows Remote Assistance genutzt wird.
Gut zu wissen: - Bei Installationsproblemen wird der Prozess angehalten und eine Fehlermeldung angezeigt. Komponenten, die erfolgreich installiert werden, bleiben gespeichert; sie müssen nicht neu installiert werden. - Studio wird nach dem Abschluss der Installation automatisch gestartet. Sie können diese Aktion während der Installation deaktivieren. - Beim Erstellen von Objekten vor, während und nach der Installation sollten Sie eindeutige Namen für jedes Objekt eingeben (z. B. Netzwerke, Gruppen, Kataloge, Ressourcen). - Nach der Installation der Komponenten in Amazon Web Services (AWS) benötigen Sie die Werte für Region, Verfügbarkeitszone, VPC-Namen, Subnetzadressen, Domännennamen, Namen der Sicherheitsgruppen und Anmeldeinformationen, um mit Studio eine Site zu erstellen.

VDA-Installationsvorbereitung

 Beschreibung
Wenn Sie einen VDA für Windows-Desktopbetriebssysteme installieren, legen Sie fest, ob die Version HDX 3D Pro installiert wird. Mit der Funktion HDX3D Pro werden Desktops und Anwendungen bereitgestellt, die am besten mit einem GPU mit Hardwarebeschleunigung funktionieren. Weitere Informationen finden Sie in der Dokumentation für HDX 3D Pro.
Legen Sie fest, wie der VDA verwendet wird. Die Standardeinstellung basiert auf der Annahme, dass Sie ein Masterimage des installierten VDA mit Maschinenerstellungsdiensten oder Provisioning Services zum Erstellen anderer virtueller Maschinen verwenden. Sie können diese Standardeinstellung überschreiben, wenn Sie den VDA auf einer vorhandenen Maschine installieren möchten.
Legen Sie fest, ob Sie Citrix Receiver für Windows (CitrixReceiver.exe) installieren. Sie können diese Standardaktion deaktivieren.
Legen Sie fest, wie Ports geöffnet werden. Standardmäßig werden die folgenden Ports automatisch geöffnet, wenn der Windows-Firewalldienst ausgeführt wird, selbst wenn die Firewall nicht aktiviert ist. Sie können diese Standardaktion deaktivieren und die Ports manuell öffnen, wenn Sie die Firewall eines Drittanbieters oder keine Firewall verwenden, oder wenn Sie die Ports einfach selbst öffnen möchten. • Controller: TCP 80, 1494, 2598, 8008 • Für die Kommunikation zwischen Benutzergeräten und virtuellen Desktops müssen Sie eingehendes TCP auf Ports 1494 und 2598 als Portausnahmen konfigurieren. Aus Sicherheitsgründen empfiehlt Citrix, diese registrierten Ports ausschließlich für das ICA-Protokoll und das Common Gateway Protocol zu verwenden. • Für die Kommunikation zwischen Controller und virtuellen Desktops müssen Sie den eingehenden Port 80 als Portausnahme konfigurieren. • Windows-Remoteunterstützung: TCP 3389 Wenn das Feature aktiviert ist, öffnet Windows diesen Port automatisch, selbst wenn Sie die Ports normalerweise manuell öffnen. • Echtzeitaudioübertragung: UDP 16500-16509 Tipp: Vollständige Portinformationen finden Sie unter CTX101810.
Legen Sie fest, wie die Speicherorte der installierten Controller angegeben werden. • Manuell durch Eingabe des vollqualifizierten Domänennamens (FQDN) des Controllers. Sie können zwar einen Controller angeben, der zurzeit nicht in der Domäne ist, VDAs können jedoch nur eine Verbindung mit einem Controller in der Domäne herstellen. Zudem kann die Verbindung nur für Controller in der Domäne getestet werden. • Mit Active Directory: wenn der Controller in der Domäne eingebunden ist. • Lassen Sie den Controller durch Maschinenerstellungsdienste angeben. • Später: Durch erneutes Ausführen des Installationsprogramms, über Citrix Richtlinien, durch Einstellen der Registrierungswerte oder mit Active Directory-Organisationseinheiten. Citrix Gruppenrichtlinieneinstellungen, die die Controller-Speicherorte angeben, überschreiben die bei der Installation festgelegten Einstellungen. Nach der Angabe des Controller-Standorts können VDAs später mit der Funktion für automatische Updates aktualisiert werden, wenn weitere Controller installiert werden. Weitere Informationen über Methoden zum Festlegen von Controllern finden Sie unter Verwalten der Delivery Controller-Umgebung.
Entscheiden Sie, ob Sie die folgenden Features verwenden möchten: • Leistung optimieren: Wenn diese Funktion aktiviert ist, wird das Optimierungstool für VDAs verwendet, die auf einer VM auf einem Hypervisor ausgeführt werden. Die VM-Optimierung umfasst das Deaktivieren von Offlinedateien, das Deaktivieren der Hintergrunddefragmentierung und die Verkleinerung der Ereignisprotokollgröße. Weitere Informationen finden Sie unter CTX125874. Aktivieren Sie diese Option nicht, wenn Sie Remote-PC-Zugriff verwenden möchten. Standard = aktiviert • Windows-Remoteunterstützung: Wenn dieses Feature aktiviert ist, wird die Windows-Remoteunterstützung mit dem Feature zur Benutzerspiegelung von Director verwendet, und Windows öffnet automatisch TCP-Port 3389 in der Firewall, selbst wenn Sie die Firewallports normalerweise manuell öffnen. Standard = aktiviert • Echtzeitaudioübertragung für Audio: Wenn diese Funktion aktiviert ist, wird UDP für Audiopakete verwendet, wodurch die Audioleistung verbessert werden kann. Standard = aktiviert • Personal vDisk: (Nur für die Installation von VDA für Windows-Desktopbetriebssysteme auf einer VM.) Wenn dieses Feature aktiviert ist, können persönliche vDisks mit einem Masterimage verwendet werden. Weitere Informationen finden Sie unter Personal vDisks. Standard = deaktiviert
Gut zu wissen: • Das Installationsprogramm erkennt automatisch das Betriebssystem und erlaubt nur die Installation des für das System geeigneten VDA-Typs: VDA für Windows-Serverbetriebssysteme oder VDA für Windows-Desktopbetriebssysteme. • Die Profilverwaltung wird bei der Installation von VDA installiert. • Wenn Sie VDA installieren, wird automatisch eine neue lokale Benutzergruppe namens "Benutzer mit direktem Zugriff" erstellt. Bei VDA für Windows-Desktopbetriebssysteme gilt diese Gruppe nur für RDP-Verbindungen; bei VDA für Windows-Serverbetriebssysteme gilt sie für ICA- und RDP-Verbindungen. • Wenn Sie einen VDA für Windows-Serverbetriebssysteme installieren, werden die Rollendienste für die Remotedesktopdienste automatisch installiert und aktiviert, wenn sie nicht bereits installiert und aktiviert sind. • Bei der Installation eines VDAs für Windows-Desktopbetriebssysteme werden auch Touch-Features (einschließlich Flicks-Anwendung) installiert. (Wenn Sie Microsoft System Center Configuration Manager verwenden, wird die Maschine nach dem Neustart als Tablet kategorisiert.) • Für Remote-PC-Zugriff-Konfigurationen ist VDA für Windows-Desktopbetriebssystem auf jedem physischen Büro-PC zu installieren, auf den die Benutzer zugreifen. • Installieren Sie diese VDA-Version für Windows-Serverbetriebssysteme nicht auf einem Server, auf dem XenApp 6.5 oder eine frühere Version installiert ist.

Installieren über die grafische Oberfläche

Hinweis: Vor Beginn einer Installation führen Sie die unter [Vorbereiten der Installation](#) beschriebenen Schritte durch.

Starten Sie die grafische Oberfläche des Installationsprogramms:

1. Laden Sie das Produktpaket herunter und entzippen Sie es. Optional können Sie die ISO-Datei auch auf DVD brennen.
2. Melden Sie sich mit einem lokalen Administratorkonto am Server an, auf dem Sie die Komponenten installieren.
3. Legen Sie die DVD in das Laufwerk ein oder stellen Sie die ISO-Datei bereit. Doppelklicken Sie auf Automatische Auswahl, wenn das Installationsprogramm nicht automatisch gestartet wird, oder

klicken Sie auf das bereitgestellte Laufwerk.

4. Wählen Sie die Komponente aus, die Sie installieren möchten:

- Wenn Sie ein Neueinsteiger sind, klicken Sie auf Delivery Controller. Anschließend können Sie den Delivery Controller installieren und wahlweise auch Studio, Director, Lizenzserver und StoreFront auf demselben Server.
- Wenn Sie bereits einige Komponenten installiert haben und Ihre Bereitstellung erweitern möchten, klicken Sie in der rechten Spalte auf die zu installierende Komponente. Diese Spalte enthält Kernkomponenten und den universellen Druckserver, den Sie auf Ihrem Druckserver installieren können.
- Klicken Sie zum Installieren eines Virtual Delivery Agent (VDA) auf den verfügbaren VDA-Eintrag. Das Installationsprogramm weiß, welcher VDA für das Betriebssystem, auf dem Sie das Installationsprogramm ausführen, der richtige ist.

Später können Sie einen installierten VDA auf folgende Weise anpassen:

1. Klicken Sie in Windows im Dialogfeld zum Hinzufügen oder Entfernen von Programmen mit der rechten Maustaste auf Citrix Virtual Delivery Agent <Versionsnummer> und wählen Sie Ändern.
2. Wählen Sie Virtual Delivery Agent-Einstellungen anpassen. Wenn das Installationsprogramm startet, können Sie die Controlleradressen ändern sowie den TCP/IP-Port für die Registrierung beim Controller (Standard = 80) festlegen und ob Portausnahmen für die Windows-Firewall automatisch geöffnet werden sollen.

Installieren an der Befehlszeile

Verwenden Sie die Befehlszeilenoberfläche für Folgendes:

- Installieren einer oder mehrerer der Kernkomponenten: Delivery Controller, Citrix Studio, Citrix Director, Lizenzserver und StoreFront.
- Installieren von Virtual Delivery Agent (VDA) auf einem Masterimage bzw. auf einer virtuellen oder physischen Maschine.
Sie können auch die Skripts anpassen, die auf dem Medium sind, und sie zum Installieren und Entfernen von VDAs in Active Directory verwenden; weitere Informationen finden Sie unter [Installieren oder Entfernen von Virtual Delivery Agents in Active Directory mit Skripts](#).
- Anpassen bereits installierter VDAs.
- Installieren eines universellen Druckservers, der Netzwerksitzungsdrucker bereitstellt. (Der Controller hat bereits die universelle Druckerserverfunktionalität; Sie müssen nur den universellen Druckserver auf den Druckservern in der Umgebung installieren.)

Mit den Optionen /remove oder /removeall können Sie zudem zuvor installierte Komponenten aus dieser Version entfernen. Weitere Informationen finden Sie unter [Entfernen von Komponenten](#).

Sie müssen der Originaladministrator sein oder verwenden Sie "Als Administrator ausführen", um den Fortschritt der Befehlsausführung und die Rückgabewerte anzuzeigen. Weitere Informationen finden Sie in der Microsoft-Befehlsdokumentation.

Wichtig: Vor Beginn einer Installation führen Sie die unter [Vorbereiten der Installation](#) beschriebenen Schritte durch.

Installieren von Kernkomponenten an der Befehlszeile

Führen Sie im Setupverzeichnis \x64\XenDesktop auf dem Medium den Befehl XenDesktopServerSetup.exe aus. Die folgende Tabelle beschreibt die Befehloptionen.

Hinweis: Geben Sie zum Installieren von XenApp die Option "/xenapp" in der Befehlszeile ein. Geben Sie zum Installieren von XenDesktop nicht die Option "/xenapp" in der Befehlszeile ein.

Option	Beschreibung
/help oder /h	Zeigt die Hilfe für Befehle an.
/quiet oder /passive	Während der Installation wird keine Benutzeroberfläche angezeigt. Der einzige Hinweis auf den Installationsvorgang ist im Windows Task-Manager. Wenn diese Option ausgelassen wird, wird die grafische Oberfläche gestartet.
/logpath Pfad	Speicherort der Protokolldateien. Der angegebene Ordner muss bereits vorhanden sein. Er wird nicht vom Installationsprogramm erstellt. Standard = "%TEMP%\Citrix\XenDesktop Installer"
/noreboot	Verhindert einen Neustart nach der Installation. (Bei den meisten Kernkomponenten ist ein Neustart in der Standardeinstellung nicht aktiviert).
/remove	Entfernt die mit /components angegebenen Kernkomponenten. Weitere Informationen über das Entfernen von Komponenten finden Sie unter Entfernen von Komponenten .
/removeall	Entfernt alle installierten Kernkomponenten. Weitere Informationen über das Entfernen von Komponenten finden Sie unter Entfernen von Komponenten .
/xenapp	Installiert XenApp. Wenn diese Option ausgelassen wird, wird XenDesktop installiert.
/configure_firewall	Öffnet alle Ports in der Windows-Firewall, die für die installierten Komponenten erforderlich sind, wenn der Windows-Firewalldienst ausgeführt wird, selbst wenn die Firewall nicht aktiviert ist. Wenn Sie die Firewall eines Drittanbieters verwenden oder keine Firewall verwenden, müssen die Ports manuell geöffnet werden.
/components component [component] ...	(erforderlich). Durch Trennzeichen getrennte Liste der zu installierenden oder zu entfernenden Komponenten. Gültige Werte: • CONTROLLER - Controller • DESKTOPSTUDIO - Studio • DESKTOPDIRECTOR - Director • LICENSESERVER - Citrix Lizenzierung • STOREFRONT - StoreFront Wenn diese Option ausgelassen wird, werden alle Komponenten installiert (bzw. entfernt, wenn die Option /remove ebenfalls angegeben ist).
/installdir Verzeichnis	Vorhandenes leeres Verzeichnis, in dem die Komponenten installiert werden. Standard: C:\Programme\Citrix
/tempdir Verzeichnis	Das Verzeichnis, das die temporären Dateien während der Installation enthält. Standard: C:\Windows\Temp
/nosql	Verhindert die Installation von Microsoft SQL Server Express auf dem Server, auf dem Sie den Controller installieren. Wenn diese Option ausgelassen wird, wird SQL Server Express installiert.

Option	Beschreibung
/no_remote_assistance	(Gilt nur für die Installation von Director.) Verhindert die Installation und Aktivierung der Windows-Remoteunterstützung.

Beispiel: Mit dem folgenden Befehl werden ein XenDesktop-Controller, Studio, die Citrix Lizenzierung und SQL Server Express auf dem Server installiert. Für die Komponentenkommunikation erforderliche Ports werden automatisch geöffnet.

```
\x64\XenDesktop Setup\XenDesktopServerSetup.exe /components controller,desktopstudio,licenseserver /configure_firewall
```

Mit dem folgenden Befehl werden ein XenApp-Controller, Studio und SQL Server Express auf dem Server installiert. Für die Komponentenkommunikation erforderliche Ports werden automatisch geöffnet.

```
\x64\XenDesktop Setup\XenDesktopServerSetup.exe /xenapp /components controller,desktopstudio /configure_firewall
```

Installieren eines VDA an der Befehlszeile

Hinweis: Weitere Informationen zur Installation einer früheren Version von Virtual Desktop Agent unter Windows XP oder Windows Vista finden Sie unter [Installieren einer früheren Version von Virtual Desktop Agent unter Windows XP oder Windows Vista](#).

Bei der Installation von VDA für die Verwendung mit Remote-PC-Zugriff geben Sie nur Optionen ein, die auf physischen Maschinen (nicht VMs oder Masterimages) gültig und für VDAs für Windows-Desktopbetriebssysteme relevant sind.

Führen Sie auf dem Produktmedium im Setupverzeichnis \x64\XenDesktop den Befehl XenDesktopVdaSetup.exe aus. Die folgende Tabelle beschreibt die Befehloptionen. Wenn nicht anders angegeben, gelten die Optionen für physische und virtuelle Maschinen sowie für VDAs für Windows-Desktopbetriebssysteme und VDAs für Windows-Serverbetriebssysteme.

Option	Beschreibung
/h oder /help	Zeigt die Hilfe für Befehle an.
/quiet oder /passive	Während der Installation wird keine Benutzeroberfläche angezeigt. Der einzige Hinweis auf den Installations- und Konfigurationsvorgang ist im Windows Task-Manager. Wenn diese Option ausgelassen wird, wird die grafische Oberfläche gestartet.
/logpath Pfad	Speicherort der Protokolldateien. Der angegebene Ordner muss bereits vorhanden sein. Er wird nicht vom Installationsprogramm erstellt. Standard = "%TEMP%\Citrix\XenDesktop Installer"
/noreboot	Verhindert einen Neustart nach der Installation. VDA steht erst nach einem Neustart vollständig zur Verfügung.
/remove	Entfernt die mit /components angegebenen Komponenten.
/removeall	Entfernt alle installierten VDA-Komponenten.
/reconfig	Passt die zuvor konfigurierten VDA-Einstellungen an, wenn der Befehl mit den Optionen /portnumber, /Controller oder /enable_hdx_ports verwendet wird. Wenn Sie diese Option ohne die Option /quiet angeben, wird die grafische Oberfläche zum Anpassen von VDA gestartet.
/portnumber port	(Gilt nur, wenn die Option /reconfig angegeben wurde.) Portnummer für die Kommunikation zwischen VDA und dem Controller. Der zuvor konfigurierte Port wird deaktiviert, es sei denn, es handelt sich um Port 80.
/components component[,component]	Durch Trennzeichen getrennte Liste der zu installierenden oder zu entfernenden Komponenten. Gültige Werte: - VDA: Installation von VDA - PLUGINS: Installation von Citrix Receiver für Windows (CitrixReceiver.exe) Wenn diese Option ausgelassen wird, werden alle Komponenten installiert.
/installdir Verzeichnis	Vorhandenes leeres Verzeichnis, in dem die Komponenten installiert werden. Standard: C:\Programme\Citrix
/tempdir Verzeichnis	Das Verzeichnis für die temporären Dateien während der Installation. (Diese Option ist bei der grafischen Oberfläche nicht verfügbar) Standard = C:\Windows\Temp.
/site_guid guid	GUID (Globally Unique Identifier) der Website Active Directory Organisationseinheit (OU). Dabei wird ein virtueller Desktop einer Site zugeordnet, wenn Active Directory für die Discovery verwendet wird (das Feature für automatische Updates ist die empfohlene und Discovery-Standardmethode). Die Site-GUID ist eine Site-Eigenschaft, die in Studio angezeigt wird. Geben Sie nicht beide Optionen, /site_guid und /controllers, an.
/controllers "controller [controller] [...]"	Durch Leerzeichen getrennte, vollqualifizierte Domännennamen (FQDNs) der Controller, mit denen VDA kommunizieren kann; von Anführungszeichen umschlossen. Geben Sie nicht beide Optionen, /site_guid und /controllers, an.
/xa_server_location url	URL des Servers für Windows Serveranwendungen.
/enable_remote_assistance	Aktiviert Windows-Remoteunterstützung für die Verwendung mit Director. Wenn Sie diese Option angeben, wird der TCP-Port 3389 durch Windows in der Firewall geöffnet, selbst wenn die Option /enable_hdx_ports ausgelassen wird.
/enable_hdx_ports	Öffnet die erforderlichen Ports für den Controller und die angegebenen Features (Windows-Remoteunterstützung, Echtzeitübertragung und Optimieren) in der Windows Firewall, wenn der Windows-Firewalldienst erkannt wird, selbst wenn die Firewall nicht aktiviert ist. Wenn Sie eine andere oder keine Firewall verwenden, müssen die Ports manuell geöffnet werden.

Option	Beschreibung
/optimize	Aktiviert die Optimierung für VDAs, die auf einer VM auf einem Hypervisor ausgeführt werden. Die VM-Optimierung umfasst das Deaktivieren von Offlinedateien, das Deaktivieren der Hintergrunddefragmentierung und die Verkleinerung der Ereignisprotokollgröße. Geben Sie diese Option für Remote-PC-Zugriff nicht an. Weitere Informationen zum Optimierungstool finden Sie unter CT X125874 .
/baseimage	(Gilt nur für die Installation von VDA für Windows-Desktopbetriebssysteme auf einer VM.) Ermöglicht das Verwenden von persönlichen vDisks mit einem Masterimage. Weitere Informationen finden Sie unter Verwalten von Personal vDisks .
/enable_hdx_3d_pro	Installiert VDA für HDX 3D Pro. Weitere Informationen finden Sie in der Dokumentation für HDX 3D Pro.
/enable_real_time_transport	Aktiviert oder deaktiviert die Verwendung von UDP für Audiopakete (Echtzeittransport für Audio). Das Aktivieren dieses Features kann die Audioleistung verbessern. Verwenden Sie die Option "/enable_hdx_ports", wenn Sie möchten, dass die UDP-Ports automatisch bei Erkennung des Windows-Firewalldiensts geöffnet werden.
/masterimage	(Gilt nur für die Installation von VDA auf einer VM.) Richtet den VDA als Masterimage ein.
/virtualmachine	(Gilt nur für die Installation von VDA auf einer VM.) Überschreibt das Erkennen einer physischen Maschine durch den Installer. Dabei werden BIOS-Informationen an die VMs weitergegeben, sodass sie als physische Maschinen erscheinen.
/nodesktopexperience	(Gilt nur für die Installation von VDA für Windows-Serverbetriebssysteme.) Verhindert das Aktivieren von Enhanced Desktop Experience. Dieses Feature wird auch über die Citrix Richtlinieneinstellung "Enhanced Desktop Experience" gesteuert.
/nocitrixwddm	(Gilt nur für Windows 7-Maschinen, die keine WDDM-Treiber enthalten.) Deaktiviert die Installation des Citrix WDDM-Treibers.
/servervdi	Installiert einen VDA für Windows-Desktopbetriebssysteme auf einem unterstützten Windows Server. Lassen Sie diese Option aus, wenn Sie einen VDA für Windows-Serverbetriebssysteme auf einem Windows Server installieren. Bevor Sie diese Option verwenden, lesen Sie den Artikel Server-VDI .
/installwithsecurebootenabled	Ermöglicht die VDA-Installation, wenn der sichere Start aktiviert ist. Wenn diese Option ausgelassen wird, gibt eine Warnung an, dass der sichere Start für eine erfolgreiche VDA-Installation deaktiviert werden muss.
/exclude "Personal vDisk","Machine Identity Service"	(Gilt nur bei Upgrades von einem älteren 7.x-VDA auf einem physischen Computer.) Schließt Personal vDisk und den Maschinenidentitätsdienst vom Upgrade aus.

Mit dem folgenden Befehl wird beispielsweise ein VDA für Windows-Desktopbetriebssysteme und Citrix Receiver am Standardspeicherort auf einer VM installiert. VDA wird als Masterimage verwendet. Der VDA registriert sich anfänglich am Controller auf dem "Contr-Main" genannten Server in der Domäne "mydomain" und verwendet Personal vDisks, das Optimierungsfeature und die Windows-Remoteunterstützung.

```
\x64\XenDesktop Setup\XenDesktopVdaSetup.exe /quiet /components vda,plugins /controllers "Contr-Main.mydomain.local" /enable_hdx_ports /optimize /masterimage /baseimage /enable_remote_assis
```

Mit dem folgenden Befehl wird VDA für Windows-Desktopbetriebssysteme und Citrix Receiver am Standardspeicherort des Büro-PCs installiert, der mit Remote-PC-Zugriff verwendet wird. Nach der Installation von VDA wird die Maschine nicht neu gestartet. Ein Neustart ist jedoch erforderlich, bevor VDA verwendet werden kann. Der VDA registriert sich anfänglich am Controller auf dem "Contr-East" genannten Server in der Domäne "mydomain" und verwendet UDP für Audiopakete. HDX-Ports werden geöffnet, wenn der Windows-Firewalldienst erkannt wird.

```
\x64\XenDesktop Setup\XenDesktopVdaSetup.exe /quiet /components vda,plugins /controllers "Contr-East.mydomain.local" /enable_hdx_ports /enable_real_time_transport /noreboot
```

Anpassen eines VDA an der Befehlszeile

Nachdem VDA installiert wurde, können Sie einige Einstellungen anpassen. Führen Sie auf dem Produktmedium vom Setupverzeichnis \x64\XenDesktop den Befehl XenDesktopVdaSetup.exe aus und legen Sie dabei eine oder mehrere der oben beschriebenen Optionen fest:

- /reconfigure: Diese Option ist zum Anpassen von VDA erforderlich
- /h oder /help
- /quiet
- /noreboot
- /controller
- /portnumber port
- /enable_hdx_ports

Installieren des universellen Druckservers an der Befehlszeile

Führen Sie einen der folgenden Befehle auf jedem Druckserver aus:

- Unterstütztes 32-Bit-Betriebssystem: Führen Sie die Datei UpsServer_x86.msi aus, die sich auf dem Citrix Installationsmedium im Verzeichnis \x86\Universal Print Server\ befindet.
- Unterstütztes 64-Bit-Betriebssystem: Führen Sie die Datei UpsServer_x64.msi aus, die sich auf dem Citrix Installationsmedium im Verzeichnis \x64\Universal Print Server\ befindet.

Installieren von VDAs mit Skripts in Active Directory

Sep 29, 2015

Das Installationsmedium enthält Beispielskripts, um Virtual Delivery Agents (VDAs) für Gruppen von Maschinen in Active Directory zu installieren, zu aktualisieren oder zu entfernen. Sie können die Skripts auch auf einzelne Maschinen anwenden und sie zum Verwalten von Masterimages einsetzen, die von den Maschinenerstellungsdiensten und Provisioning Services verwendet werden.

Erforderliche Zugriffsberechtigungen:

- Für die Skripts ist Lesezugriff für "Jeder" auf der Netzwerkfreigabe erforderlich, auf der der VDA-Installationsbefehl (XenDesktopVdaSetup.exe) ist.
- Die Protokolldetails werden auf jeder lokalen Maschine gespeichert. Sollen die Ergebnisse auch zentral zur Überprüfung und Analyse protokolliert werden, benötigen die Skripts Lese- und Schreibzugriff auf der Netzwerkfreigabe für "Jeder".

Um die Ergebnisse der Skriptausführung zu überprüfen, müssen Sie die zentrale Protokollfreigabe untersuchen. Erfasst werden das Skriptprotokoll, das Installationsprogrammprotokoll und die MSI-Installationsprotokolle. Jeder Installations- oder Deinstallationsvorgang wird in einem Ordner mit Zeitstempel aufgezeichnet. Am Präfix "PASS" oder "FAIL" im Ordnername ist ersichtlich, ob der Vorgang erfolgreich war oder nicht. Sie können herkömmliche Verzeichnissuchprogramme verwenden, um schnell eine fehlerhafte Installation oder Deinstallation im zentralen Protokoll zu finden, statt lokal auf den Zielmaschinen zu suchen. Weitere Informationen finden Sie weiter unten im Abschnitt "Problembehandlung".

Wichtig: Vor Beginn einer Installation führen Sie die unter [Vorbereiten der Installation](#) beschriebenen Schritte durch. Installieren oder Aktualisieren von VDAs mit dem Skript

1. Besorgen Sie sich das Beispielskript InstallVDA.bat aus \Support\AdDeploy\ auf dem Installationsmedium. Citrix empfiehlt, dass Sie ein Backup der ursprünglichen Skriptdatei anlegen, bevor Sie sie ändern.
2. Bearbeiten Sie das Skript:
 - Geben Sie die Version des zu installierenden VDA an: SET DESIREDVERSION. Beispielsweise kann die Version 7 als 7.0 angegeben werden. Der vollständige Wert befindet sich auf dem Installationsmedium in der Datei ProductVersion.txt (z. B. 7.0.0.3018); eine vollständige Übereinstimmung ist jedoch nicht erforderlich.
 - Geben Sie die Netzwerkfreigabe an, von der das Installationsprogramm aufgerufen wird. Verweisen Sie auf den Stamm des Layouts (der höchste Punkt der Struktur): die geeignete Version des Installationsprogramms (32 Bit oder 64 Bit) wird automatisch aufgerufen, wenn das Skript ausgeführt wird. Beispiel: SET DEPLOYSHARE=\\filesrv1\share1.
 - Geben Sie optional einen Netzwerkfreigabeort zum Speichern der zentralen Protokolle an. Beispiel: SET LOGSHARE=\\filesrv1\log1).
 - Geben Sie die VDA-Konfigurationsoptionen wie unter [Installieren an der Befehlszeile](#) beschrieben ein. Die Optionen /quiet und /noreboot sind standardmäßig im Skript enthalten und sind erforderlich: SET COMMANDLINEOPTIONS=/QUIET /NOREBOOT.
3. Weisen Sie mit den Startskripts für Gruppenrichtlinien das Skript der Organisationseinheit in Active Directory zu, in der die Maschinen sind. Diese Organisationseinheit sollte nur Maschinen enthalten, auf denen Sie VDA installieren möchten. Wenn die Maschinen in der Organisationseinheit neu gestartet werden, wird das Skript auf allen ausgeführt und VDA auf jeder Maschine installiert, deren Betriebssystem unterstützt wird.

Entfernen von VDAs mit dem Skript

1. Besorgen Sie sich das Beispielskript UninstallVDA.bat aus \Support\AdDeploy\ auf dem Installationsmedium. Citrix empfiehlt, dass Sie ein Backup der ursprünglichen Skriptdatei anlegen, bevor Sie sie ändern.
2. Bearbeiten Sie das Skript.

- Geben Sie die Version des zu entfernenden VDAs an: SET CHECK_VDA_VERSION. Beispielsweise kann die Version 7 als 7.0 angegeben werden. Der vollständige Wert befindet sich auf dem Installationsmedium in der Datei ProductVersion.txt (z. B. 7.0.0.3018); eine vollständige Übereinstimmung ist jedoch nicht erforderlich.
 - Geben Sie optional einen Netzwerkfreigabeort zum Speichern der zentralen Protokolle an.
3. Weisen Sie mit den Startskripts für Gruppenrichtlinien das Skript der Organisationseinheit in Active Directory zu, in der die Maschinen sind. Diese Organisationseinheit sollte nur Maschinen enthalten, von denen Sie VDA entfernen möchten. Wenn die Maschinen in der Organisationseinheit neu gestartet werden, wird das Skript auf allen ausgeführt und VDA auf jeder Maschine entfernt.

Problembehandlung

Das Skript generiert interne Protokolldateien, die den Skriptausführungsverlauf beschreiben. Das Skript kopiert das Protokoll Kickoff_VDA_Startup_Skript innerhalb von Sekunden nachdem die Bereitstellung auf der Maschine gestartet wurde in die zentrale Protokollfreigabe, sodass Sie den Gesamtvorgang überprüfen können. Wenn dieses Protokoll nicht wie vorgesehen in die Protokollfreigabe kopiert wird, können Sie zur weiteren Fehlerbehebung die lokale Maschine untersuchen, denn durch das Skript werden zwei Debugging-Protokolldateien im Ordner %temp% auf jeder Maschine gespeichert:

- Kickoff_VDA_Startup_Skript_.log
- VDA_Install_ProcessLog_.log

Überprüfen Sie den Inhalt dieser Protokolle, um Folgendes für das Skript sicherzustellen:

- Es wird wie erwartet ausgeführt.
- Das Zielbetriebssystem wird korrekt erkannt.
- Der Verweis auf ROOT von DEPLOYSHARE ist korrekt konfiguriert (enthält die Datei "AutoSelect.exe").
- Die Authentifizierung bei den Freigaben DEPLOYSHARE und LOG ist möglich.

Erstellen einer Site

Sep 29, 2015

Eine Site ist der Name, den Sie einer Produktbereitstellung geben. Sie umfasst die Delivery Controller und die anderen Kernkomponenten, VDAs, Verbindungen mit virtuellen Ressourcen (falls verwendet) sowie die Maschinenkataloge und Bereitstellungsgruppen, die Sie erstellen und verwalten. Eine Site muss nicht unbedingt einem geografischen Standort entsprechen, obwohl dies der Fall sein kann. Sie erstellen die Site nach der Installation der Komponenten und bevor Sie Maschinenkataloge und Bereitstellungsgruppen erstellen.

Vorbereiten

In der folgenden Tabelle finden Sie die Aufgaben, die Sie abschließen müssen, und Punkte, die Sie berücksichtigen sollten, bevor Sie den Assistenten für die Siteerstellung in Studio starten.

✓ Beschreibung			
Legen Sie fest, welchen Sitetyp Sie erstellen möchten: • Anwendungs- und Desktopbereitstellungssite: Wenn Sie eine Anwendungs- und Desktopbereitstellungssite erstellen möchten, haben Sie die Wahl zwischen einer vollständigen Bereitstellung (empfohlen) oder einer leeren Site. (Leere Sites sind nur teilweise konfiguriert und werden normalerweise von erfahrenen Benutzern erstellt.) • Remote-PC-Zugriff-Site: Ermöglicht bestimmten Benutzern den Remotezugriff auf ihre Büro-PCs über eine sichere Verbindung. Wenn Sie das Wake-On-LAN-Feature von Remote-PC-Zugriff verwenden möchten, lesen Sie den Abschnitt Remote-PC-Zugriff führen Sie die unter Microsoft System Center Configuration Manager und Remote-PC-Zugriff-Wake-On-LAN beschriebenen Aufgaben durch. Wenn Sie zu diesem Zeitpunkt eine Bereitstellung für die Anwendungs- und Desktopbereitstellung erstellen, können Sie eine Remote-PC-Zugriff-Bereitstellung später hinzufügen. Ebenso können Sie einer Remote-PC-Zugriff-Bereitstellung später eine vollständige Bereitstellung hinzufügen.			
Die Siteerstellung umfasst das Erstellen der Sitekonfigurationsdatenbank. Stellen Sie sicher, dass die SQL Server-Software installiert ist, bevor Sie eine Site erstellen. Zum Erstellen der Datenbank müssen Sie ein lokaler Administrator und ein Domänenbenutzer sein. Sie müssen zudem entweder SQL Server-Berechtigungen haben oder Sie können Skripts generieren, die dann vom Datenbankadministrator ausgeführt werden. • Berechtigungen: Sie benötigen die folgenden Berechtigungen zum Einrichten der Datenbank. Die Berechtigungen können explizit konfiguriert oder über eine Active Directory-Gruppenmitgliedschaft erworben werden:			
Vorgang	Zweck	Serverrolle	Datenbankrolle
Datenbank erstellen	Erstellen einer geeigneten leeren Datenbank	dbcreator	
Schema erstellen	Erstellen aller dienstspezifischen Schemas und Hinzufügen des ersten Controllers zur Site	securityadmin *	db_owner
Controller hinzufügen	Hinzufügen eines weiteren Controllers (zusätzlich zum ersten) zur Site	securityadmin *	db_owner

 Beschreibung	Zweck	Serverrolle securityadmin*	Datenbankrolle
Controller hinzufügen (Spiegelungsserver)	Hinzufügen einer Controller-Anmeldung zu dem Datenbankserver, der derzeit die Spiegelrolle einer gespiegelten Datenbank hat		
Schema aktualisieren	Anwenden von Aktualisierungen oder Hotfixes auf das Schema		db_owner
* Zwar ist die securityadmin-Serverrolle technisch restriktiver als die sysadmin-Serverrolle, aber in der Praxis ist sie als gleichwertig anzusehen. Wenn Sie diese Vorgänge mit Studio ausführen, muss das Benutzerkonto Mitglied der sysadmin-Serverrolle sein.			
Wenn Ihre Studio-Anmeldeinformationen diese Berechtigungen nicht umfassen, werden Sie aufgefordert, Benutzeranmeldeinformationen für SQL Server einzugeben. • Skripts: Wenn der Datenbankserver gesperrt ist und Sie nicht über die erforderlichen SQL Server-Berechtigungen verfügen, kann der Assistent für die Siteerstellung zwei Datenbankskripts erstellen, eines zum Einrichten der Datenbank und das zweite zur Verwendung in einer Spiegelungsumgebung. Nachdem Sie die Skripterstellung angefordert haben, geben Sie die erstellten Skripts dem Datenbankadministrator (oder einer Person mit den erforderlichen SQL Server-Berechtigungen) zur Ausführung auf dem Datenbankserver und der gespiegelten Datenbank (falls erforderlich). Nachdem das Skript ausgeführt und die Datenbank erfolgreich erstellt wurde, können Sie die Erstellung der Site abschließen.			
Erwägen Sie, ob Sie das 30-tägige kostenlose Probeabo, bei dem die Lizenzdateien später hinzugefügt werden können, oder ob Sie vorhandene Lizenzen verwenden. Sie können Lizenzdateien im Assistenten für die Siteerstellung hinzufügen oder herunterladen.			
Konfigurieren Sie die Umgebung der Virtualisierungsressource (Hostumgebung). Wenn Sie XenServer verwenden: • Weitere Informationen finden Sie in der XenServer-Dokumentation. • Sie müssen die Anmeldeinformationen eines VM-Hauptadministrators oder eines höherrangigen Benutzers eingeben. • Citrix empfiehlt die Verwendung von HTTPS zur Sicherung der Kommunikation mit XenServer. Um HTTPS zu verwenden, müssen Sie das auf XenServer installierte Standard-SSL-Zertifikat, durch ein Zertifikat einer vertrauenswürdigen Zertifizierungsstelle ersetzen; siehe CTX128656. • Sie können hohe Verfügbarkeit konfigurieren, wenn diese auf XenServer aktiviert ist. • Citrix empfiehlt, dass Sie alle Server im Pool auswählen, um die Kommunikation mit XenServer zu ermöglichen, wenn der Poolmaster ausfällt. • Sie können auch einen GPU-Typ und eine GPU-Gruppe oder die Passthrough-Authentifizierung auswählen, wenn der XenServer vGPU unterstützt. Es wird angezeigt, ob die Auswahl dedizierte GPU-Ressourcen umfasst. Bei der Verwendung von VMware konsultieren Sie die Produktdokumentation sowie Verwalten von virtuellen Maschinen mit VMware. Bei Verwendung von Hyper-V konsultieren Sie die Produktdokumentation und Verwalten von virtuellen Maschinen			



mit Microsoft System Center Virtual Machine Manager.

Beschreibung

Entscheiden Sie, ob Sie Maschinenerstellungsdienste (MCS) oder andere Tools zum Erstellen von VMs auf den Virtualisierungsressourcen verwenden möchten.

Entscheiden Sie, ob Sie freigegebenen oder lokalen Speicher verwenden. Freigegebener Speicher ist über das Netzwerk verfügbar. Wenn Sie einen freigegebenen Speicher verwenden, können Sie die Verwendung von IntelliCache aktivieren, um die Last auf dem Speichergerät zu reduzieren. Weitere Informationen über IntelliCache finden Sie unter [Verwenden von IntelliCache](#).

Entscheiden Sie, ob persönliche vDisks verwendet werden und ob diese freigegebenen oder lokalen Speicher verwenden. Persönliche vDisks können den gleichen oder anderen Speicher als die VMs verwenden. Weitere Informationen finden Sie unter [Personal vDisks](#).

Wenn Sie Produktkomponenten in einer Cloudumgebung installiert haben, benötigen Sie beim Konfigurieren der ersten Verbindung den API-Schlüssel und die Werte des geheimen Schlüssels. Sie können die Schlüsseldatei mit diesen Werten aus AWS oder CloudPlatform exportieren und anschließend in den Assistenten für die Siteerstellung importieren.

Wenn Sie eine Site für eine Cloudbereitstellung erstellen, benötigen Sie zudem die in AWS konfigurierten Werte für Region, Availability Zone, VPC-Name, Subnetzadressen, Domänenname, Namen der Sicherheitsgruppen und Anmeldeinformationen.

Entscheiden Sie, ob Sie App-V-Veröffentlichung verwenden möchten, und konfigurieren Sie ggf. die entsprechenden Ressourcen. Weitere Informationen finden Sie unter [Microsoft Application Virtualization](#).

Gut zu wissen:

- Für das Erstellen einer Remote-PC-Zugriff-Site gilt Folgendes:
 - Es werden automatisch ein Maschinenkatalog unter dem Namen "Remote-PC-Zugriff-Maschinen" und eine Bereitstellungsgruppe unter dem Namen "Remote-PC-Zugriff-Desktops" erstellt.
 - Sie müssen Benutzer bzw. Benutzergruppen angeben. Es gibt keine Standardaktion zum automatischen Hinzufügen aller Benutzer.
 - Sie können das Wake-On-LAN-Feature (Energieverwaltung) aktivieren und die Adresse und Anmeldeinformationen für Microsoft System Center Configuration Manager (ConfigMgr) sowie einen Verbindungsnamen angeben.
- Der Benutzer, der eine Site erstellt, ist Volladministrator dieser Site. Weitere Informationen finden Sie unter [Delegierte Administration](#).
- Wenn eine leere Datenbank erstellt wird, hat sie Standardattribute mit folgenden Ausnahmen:
 - Die Sortierreihenfolge wird auf Latin1_General_100_CI_AS_KS festgelegt, wobei Latin1_General je nach Land variiert, z. B. Japanese_100_CI_AS_KS). Wenn diese Sortiereinstellung bei der Datenbankerstellung nicht angegeben wird, schlägt die anschließende Erstellung der Dienstschemas in der Datenbank fehl und ein Fehler wie "Schema erfordert eine Datenbank, bei der Groß-/Kleinschreibung nicht beachtet wird." wird angezeigt. (Wenn eine Datenbank manuell erstellt wird, kann eine beliebige Sortierreihenfolge verwendet werden, wenn sie Groß- und Kleinschreibung sowie Akzente und japanische Schriftzeichen erkennt. Der Name der Sortierreihenfolge endet normalerweise mit _CI_AS_KS.)
 - Der Wiederherstellungsmodus ist auf "Einfach" eingestellt. Für die Verwendung als gespiegelte Datenbank ändern Sie den Wiederherstellungsmodus zu "Vollständig".

-  Wenn Sie eine Sitekonfigurationsdatenbank erstellen, speichert diese auch Konfigurationsänderungen, die vom Konfigurationsprotokollierungsdienst aufgezeichnet werden, sowie Trend- und Leistungsdaten, die vom Überwachungsdienst verwendet und in Citrix Director angezeigt werden. Wenn Sie diese Features verwenden und Daten von mehr als sieben Tagen speichern, empfiehlt Citrix, nach der Erstellung einer Site die Datenbanken für die Konfigurationsprotokollierung und die Überwachung (sekundäre Datenbanken) an unterschiedlichen Orten zu speichern. Weitere Informationen finden Sie unter [Ändern des Speicherorts der sekundären Datenbanken](#).

Erstellen

Starten Sie Studio, falls es nicht geöffnet ist. Geben Sie nach dem Erstellen einer Site im mittleren Bereich Folgendes an:

- Den Sitetyp und den Namen der Site.
- Datenbankinformationen. Wenn Sie während der Controller-Installation die Installation der SQL Server Express-Standarddatenbank angefordert haben, werden einige Informationen bereits bereitgestellt. Wenn Sie einen Datenbankserver verwenden, der auf einem anderen Server installiert ist, geben Sie den Datenbankserver und -namen ein:

Datenbanktyp	Eingabe	Auswirkungen dieser Datenbankkonfiguration
Eigenständig oder gespiegelt	servername	Die Standardinstanz wird verwendet und SQL Server verwendet den Standardport.
	servername\INSTANCENAME	Eine benannte Instanz wird verwendet und SQL Server verwendet den Standardport.
	servername,Portnummer	Die Standardinstanz wird verwendet und SQL Server verwendet einen benutzerdefinierten Port. (Das Komma ist erforderlich.)
Sonstiges	cluster-name	Eine geclusterte Datenbank.
	availability-group-listener	Eine stets aktivierte Datenbank.

Nachdem Sie auf Weiter klicken und die Benachrichtigung erhalten, dass die Dienste keine Verbindung zu einer Datenbank herstellen konnten, geben Sie an, dass sie von Studio erstellt werden soll. Wenn Sie keine Berechtigung zum Bearbeiten der Datenbank haben, verwenden Sie Datenbankskript erstellen. Die Skripts müssen ausgeführt werden, bevor die Site vollständig erstellt wird.

- Lizenzserveradresse wie folgt: Name:[Port], wobei Name ein vollqualifizierter Domänenname (FQDN), eine NetBIOS- oder IP-Adresse ist. Das empfohlene Format ist FQDN. Wenn Sie die Portnummer auslassen, ist der Standardport 27000. Sie können erst fortfahren, wenn eine erfolgreiche Verbindung zum Lizenzserver hergestellt wurde.
- (Nur Remote-PC-Zugriff-Sites.) Energieverwaltungsinformationen, einschließlich ConfigMgr-Verbindungsinformationen.
- Informationen für die Verbindung mit der Virtualisierungsressource und Speicherinformationen. Wenn Sie keine Ressource verwenden oder Studio für die Verwaltung von Benutzerdesktops nutzen, die auf dedizierten Blade-PCs gehostet werden, wählen Sie den Verbindungstyp Keine.
- Angaben zum App-V-Verwaltungsserver und App-V-Veröffentlichungsserver.
- (Nur Remote-PC-Zugriff-Sites.) Benutzer- und Maschinenkonteninformationen.
 - Benutzerinformationen. Klicken Sie auf Benutzer hinzufügen. Wählen Sie Benutzer und Benutzergruppen aus und klicken Sie dann auf Benutzer hinzufügen.
 - Informationen zu Maschinenkonten Klicken Sie auf Maschinenkonten hinzufügen. Wählen Sie die Maschinenkonten

aus und klicken Sie dann auf Maschinenkonten hinzufügen. Klicken Sie auf Organisationseinheiten hinzufügen. Wählen Sie die Domäne und die Organisationseinheiten und geben Sie an, ob Elemente in Unterordnern eingeschlossen werden. Klicken Sie auf Organisationseinheiten hinzufügen.

Testen einer Sitekonfiguration

Konfigurations- und Umgebungstests werden automatisch ausgeführt, wenn Sie eine Site erstellen. Sie können einen HTML-Bericht der Ergebnisse anzeigen. Sie können diese Tests auch bei Bedarf ausführen:

1. Klicken Sie in Studio oben im linken Bereich auf den Eintrag Studio ().
2. Klicken Sie im mittleren Bereich auf Konfiguration testen.

Ändern des Speicherorts der sekundären Datenbanken

Sep 29, 2015

Standardmäßig befinden sich die Konfigurationsprotokollierungsdatenbank und die Überwachungsdatenbank (die sekundären Datenbanken) auf demselben Server wie die Sitekonfigurationsdatenbank. Zunächst haben alle drei Datenbanken denselben Namen. Citrix empfiehlt, den Speicherort der sekundären Datenbanken nach dem Erstellen einer Site zu ändern. Sie können die Konfigurationsprotokollierungs- und die Überwachungsdatenbanken auf einem Server oder auf unterschiedlichen Servern hosten. Die Backupstrategie kann für jede Datenbank anders sein.

Berücksichtigen Sie Folgendes, wenn Sie den Speicherort der Konfigurationsprotokollierungs- oder Überwachungsdatenbank ändern:

- Die Daten werden nicht aus der bestehenden Datenbank in die neue Datenbank importiert.
- Die Protokolle beider Datenbanken können beim Abrufen von Protokollen nicht aggregiert werden.
- Der erste Protokolleintrag in der neuen Datenbank gibt an, dass eine Datenbankänderung stattfand, die vorherige Datenbank wird jedoch nicht angegeben.

Bevor Sie den Speicherort der Konfigurationsprotokollierungs- oder der Überwachungsdatenbank ändern, müssen Sie eine unterstützte Version von Microsoft SQL Server auf dem Server installieren, auf dem die Datenbank gespeichert wird. Richten Sie je nach Bedarf Spiegel, Cluster oder andere unterstützte Redundanzinfrastrukturen ein.

Sie können den Speicherort der Konfigurationsprotokollierungsdatenbank nicht ändern, wenn die verbindliche Protokollierung aktiviert ist.

Hinweis: Sie können mit dieser Methode den Speicherort der Sitekonfigurationsdatenbank nicht ändern.

1. Wählen Sie in Studio im linken Bereich Konfiguration aus. Die Namen und Adressen der drei Datenbanken werden aufgelistet sowie die Spiegelserveradressen, falls konfiguriert.
2. Wählen Sie die Datenbank aus, für die Sie einen neuen Speicherort angeben möchten, und klicken Sie dann im Aktionsbereich auf Datenbank ändern.
3. Geben Sie im Dialogfeld Datenbank ändern den Speicherort des Servers mit der neuen SQL Serverinstallation ein (in einem der Formate aus der folgenden Tabelle) sowie den Namen der Datenbank.

Datenbanktyp	Eingabe	Auswirkungen dieser Datenbankkonfiguration
Eigenständig oder gespiegelt	servername	Die Standardinstanz wird verwendet und SQL Server verwendet den Standardport.
	Servername\Instanzname	Eine benannte Instanz wird verwendet und SQL Server verwendet den Standardport.
	servername,Portnummer	Die Standardinstanz wird verwendet und SQL Server verwendet einen benutzerdefinierten Port. (Das Komma ist erforderlich.)
Sonstiges	cluster-name	Eine geclusterte Datenbank.
	availability-group-listener	Eine stets aktivierte Datenbank.

4. Wenn Sie die Datenbank von Studio erstellen lassen möchten, klicken Sie auf OK. Wenn Sie dazu aufgefordert werden, klicken Sie auf OK und Studio erstellt die Datenbank automatisch. Studio versucht, mit den Anmeldeinformationen des aktuellen Studio-Benutzers auf die Datenbank zuzugreifen; wenn dies fehlschlägt, werden Sie zur Eingabe der Anmeldeinformationen des Datenbankbenutzers aufgefordert. Das Datenbankschema wird dann von Studio in die Datenbank hochgeladen. (Die Anmeldeinformationen werden nur für den Zeitraum der Datenbankerstellung gespeichert.)
5. Wenn Sie die Datenbank manuell erstellen möchten, klicken Sie auf Skript erstellen. Die generierten Skripts enthalten Anweisungen, wie Sie die Datenbank und ggf. die Spiegeldatenbank manuell erstellen. Stellen Sie vor dem Hochladen des Schemas sicher, dass die Datenbank leer ist und dass mindestens ein Benutzer Zugriffs- bzw. Änderungsberechtigung für die Datenbank hat.

Installieren von VDA auf älteren Systemen

Sep 29, 2015

Die mit dieser Versionsnummer gelieferten Virtual Delivery Agents (VDAs) werden unter Windows XP und Windows Vista nicht unterstützt. Darüber hinaus werden einige Features dieses Releases unter diesen Betriebssystemen nicht unterstützt. Zur Nutzung der vollständigen Funktionalität dieses Releases empfiehlt Citrix, dass Sie Ihr Windows XP- oder Windows Vista-System durch Windows 7 oder Windows 8 ersetzen und dann Virtual Delivery Agent aus diesem Release installieren.

In den Fällen, in denen eine Vorversion von Virtual Desktop Agent zum Erstellen eines Maschinenkatalogs und einer Bereitstellungsgruppe zum Hosten von Windows XP- oder Windows Vista-Clients oder zum Erstellen eines Masterimages für die Bereitstellung von Windows XP- oder Windows Vista-Desktops erforderlich ist, stehen auf dem Medium für dieses Release Vorversionen der Virtual Desktop Agent-Software zur Verfügung, die auf diesen Systemen unterstützt werden (Version 5.6 FP1). Wenn das Installationsprogramm ein Windows XP- oder Windows Vista-System erkennt, wird ein anderes Installationsprogramm gestartet, mit dem die Version 5.6 FP1 von Virtual Desktop Agent bereitgestellt wird.

- Die Kernkomponenten (z. B. Controller, Studio Director, StoreFront, Citrix Lizenzserver) können nicht auf einem Windows XP- oder Windows Vista-System installiert werden.
- Remote-PC-Zugriff wird auf Windows Vista-Systemen nicht unterstützt.

Hinweis: Wenn Sie die frühere Version eines Virtual Desktop Agent auf einer XP-Maschine (32 Bit) installieren, verwenden Sie entweder das Installationsmedium lokal oder kopieren Sie es auf das lokale Laufwerk. Versuchen Sie nicht, die frühere Version von einer Netzwerkfreigabe oder einem zugeordneten Laufwerk aus zu installieren.

1. Starten Sie auf der Maschine, auf der Windows XP oder Windows Vista ausgeführt wird, das Installationsprogramm für dieses Release und wählen Sie Virtual Delivery Agent für Windows-Desktopbetriebssysteme.
2. Wenn das Installationsprogramm für Version 5.6 FP1 von Virtual Desktop Agent gestartet wird, wählen Sie Erweitert und folgen Sie den Anweisungen auf dem Bildschirm. Detaillierte Informationen darüber, was Sie beim Installieren der früheren Version angeben müssen, finden Sie unter Installieren von Virtual Desktop Agent in der archivierten [Dokumentation](#) zu XenDesktop 5.
3. Installieren Sie anschließend alle zusätzlichen Hotfixes für dieses Release.

Führen Sie im Verzeichnis \x64\XP XenDesktop Setup oder \x86\XP XenDesktop Setup auf dem Installationsmedium den Befehl XenDesktopVdaSetup.exe aus. Detaillierte Informationen darüber, was Sie beim Installieren der früheren Version angeben müssen, finden Sie unter XenDesktopVdaSetup.exe in der archivierten [Dokumentation](#) zu XenDesktop 5.

Installieren Sie anschließend alle zusätzlichen Hotfixes für dieses Release.

Entfernen von Komponenten

Sep 29, 2015

Zum Entfernen von Komponenten empfiehlt Citrix die Verwendung der Windows-Funktion zum Entfernen oder Ändern von Programmen. Alternativ können Sie Komponenten über die Befehlszeile oder mit einem auf dem Installationsmedium enthaltenen Skript entfernen.

Beim Entfernen von Komponenten werden keine Voraussetzungen entfernt und keine Firewall-Einstellungen geändert. Wenn Sie einen Controller entfernen, werden die SQL-Serversoftware und die Datenbanken nicht entfernt.

Bevor Sie einen Controller entfernen, müssen Sie ihn aus der Site entfernen; siehe [Entfernen eines Controllers](#). Vor dem Entfernen von Studio oder Director empfiehlt Citrix das Schließen dieser Anwendungen.

Wenn Sie einen Controller von einer früheren Bereitstellung mit dem Web Interface aktualisiert haben, müssen Sie zuerst die Webinterface-Komponente separat entfernen. Das Webinterface kann nicht mit dem Installationsprogramm entfernt werden.

Gehen Sie mit der Windows-Funktion zum Entfernen oder Ändern von Programmen wie folgt vor:

- Zum Entfernen eines Controllers, von Studio, Director, eines Lizenzservers oder von StoreFront wählen Sie Citrix XenApp oder Citrix XenDesktop, klicken Sie mit der rechten Maustaste und wählen Sie Deinstallieren. Das Installationsprogramm wird gestartet und Sie können die zu entfernenden Komponenten markieren.
Alternativ können Sie StoreFront entfernen, indem Sie mit der rechten Maustaste auf Citrix StoreFront klicken und dann Deinstallieren auswählen.
- Klicken Sie zum Entfernen eines VDAs auf Citrix Virtual Delivery Agent, klicken Sie mit der rechten Maustaste und wählen Sie Deinstallieren. Das Installationsprogramm wird gestartet und Sie können die zu entfernenden Komponenten markieren.
- Zum Entfernen des universellen Druckservers wählen Sie Citrix Universeller Druckserver, klicken Sie mit der rechten Maustaste und wählen Sie Deinstallieren.

Führen Sie auf dem Installationsmedium im Setupverzeichnis `\x64\XenDesktop` den Befehl `XenDesktopServerSetup.exe` aus.

- Zum Entfernen einer oder mehrerer Kernkomponenten verwenden Sie die Optionen `/remove` und `/components`.
- Zum Entfernen aller Kernkomponenten verwenden Sie die Option `/removeall`.

Informationen zu Befehl und Parametern finden Sie unter [Installieren an der Befehlszeile](#).

Mit dem folgenden Befehl wird beispielsweise Studio entfernt:

```
\x64\XenDesktop Setup\XenDesktopServerSetup.exe /remove /components studio
```

Führen Sie auf dem Installationsmedium im Setupverzeichnis `\x64\XenDesktop` den Befehl `XenDesktopServerSetup.exe` aus.

- Zum Entfernen einer oder mehrerer Kernkomponenten verwenden Sie die Optionen `/remove` und `/components`.
- Zum Entfernen aller Kernkomponenten verwenden Sie die Option `/removeall`.

Informationen zu Befehl und Parametern finden Sie unter [Installieren an der Befehlszeile](#).

Mit dem folgenden Befehl werden beispielsweise der VDA und Receiver entfernt:

```
\x64\XenDesktop Setup\XenDesktopVdaSetup.exe /removeall
```

Informationen zum Entfernen von VDAs mit einem Skript in Active Directory finden Sie unter [Installieren oder Entfernen von Virtual Delivery Agents in Active Directory mit Skripts](#).

Upgrade einer vorhandenen Umgebung

Sep 29, 2015

Sie können das Upgrade oder die Migration von XenApp auf XenDesktop nicht mit den automatisierten Migrations- oder Updatetools durchführen und Sie können nicht von XenDesktop auf XenApp aktualisieren oder migrieren.

Wenn Sie XenApp 6.5-Bereitstellung oder früher haben, erstellen Sie eine neue Umgebung.

Wenn Sie eine XenDesktop 4.x-Bereitstellung haben, erstellen Sie zunächst eine neue Umgebung und übertragen Sie dann mit dem Migrationstool Daten und Einstellungen zur neuen Site. Weitere Informationen finden Sie unter [Migrieren von XenDesktop 4](#).

Mit einem direkten Upgrade können Sie ein Upgrade von XenDesktop 5 oder XenDesktop 7 auf die neueste Version schnell und einfach ohne Unterbrechung oder Neuplanung der XenDesktop-Bereitstellung durchführen. Nach dem Upgrade können Benutzer sofort von den Vorteilen der neuen Features dieser Releases profitieren.

Detaillierte Upgradeinformationen finden Sie unter:

- [Upgrade von XenDesktop 5](#)
- [Upgrade von XenDesktop 7](#)

Beachten Sie Folgendes:

- Führen Sie keine Upgrades von Technology Preview- oder Early Release-Versionen durch.
- Sie können kein Upgrade früherer XenApp-Versionen auf die aktuelle Version durchführen.
- Sie können kein Upgrade von XenApp auf XenDesktop durchführen.

Upgrade von XenDesktop 5

Sep 29, 2015

Vor dem Upgrade einer XenDesktop 5-Site sollten Sie sich mit den neuen Konzepten, den zu aktualisierenden Komponenten sowie deren Versionen und der Reihenfolge für ein erfolgreiches Upgrade vertraut machen.

Neue Konzepte und Features

Im Vergleich zu früheren Versionen enthält XenDesktop 7.x zahlreiche neue Features und Konzepte.

- Informationen zu neuen Konzepten, Terminologie und Systemanforderungen finden Sie unter [Info über XenDesktop](#) und [Informationen für Administratoren von früheren XenDesktop-Versionen](#).
- Informationen für XenApp-Administratoren finden Sie unter [Wichtige Informationen für XenApp-Administratoren](#).

Es gibt kein Upgrade von XenApp auf XenDesktop.

Die Studio-Verwaltung und Director können nur XenDesktop 7.x- und XenApp 7.5-Sites überwachen und verwalten. Die Überwachungs- und Verwaltungstools unterstützen keine älteren Versionen von XenDesktop und XenApp.

Wenn Sie weiterhin Bereitstellungen älterer Versionen von XenApp oder XenDesktop ausführen, empfiehlt Citrix, dass Sie diese parallel mit der XenDesktop 7.x-Site ausführen und die Verwaltungskonsolen für jedes Release für diese Site ausführen.

Beispiel: Wenn Sie in einer gemischten Umgebung weiterhin mit Desktop Director 2.1 XenApp 6.5 überwachen, stellen Sie sicher, dass Desktop Director 2.1 auf einem anderen Server als Director 7 installiert ist.

Sites mit Controllern der Version 5.x und Virtual Delivery Agents (VDAs) der Version 7.x sollten in diesem Zustand nur vorübergehend verbleiben. Im Idealfall sollten Sie das Upgrade aller Komponenten so schnell wie möglich durchführen.

Citrix empfiehlt die Verwendung von StoreFront, um die Anwendungen und Desktops der verschiedenen Versionen von XenApp und XenDesktop zu aggregieren. Weitere Informationen finden Sie in der Dokumentation zu StoreFront.

Folgende Komponenten müssen aktualisiert werden:

- Lizenzserver und Lizenzen
Hinweis: Nach einem Upgrade auf XenApp Platinum Edition für die Aktivierung von Platinum-Features wie Provisioning Services (PVS) ist es möglich, dass XenDesktop die Lizenzen nicht wie erwartet verwenden kann. Die Ursache ist, dass Studio keine XenApp Platinum-Lizenzen erkennen kann. Damit XenApp Platinum-Lizenzen verwendet werden können, ändern Sie mit dem PowerShell-SDK die Lizenzeinstellungen wie unter [Use XenApp Platinum licenses with XenDesktop 7](#) erläutert.
- Virtual Desktop Agents für Desktopbetriebssystemmaschinen (Windows-Desktop), die auf XenDesktop 7.x-VDAs aktualisiert werden
- Delivery Controller und andere Infrastrukturkomponenten, z. B. Director
- Vorhandene Datenbank

Wichtig: Bevor Sie ein Upgrade durchführen, erstellen Sie ein Backup Ihrer Datenbank wie unter [How to backup and restore your XenDesktop Database](#) beschrieben.

Empfehlungen zum VDA-Upgrade finden Sie unter [XenDesktop 5-Upgradekomponenten](#).

VDAs unter Windows XP oder Windows Vista

Einige XenDesktop 7.x-Features werden unter Windows XP oder Windows Vista nicht unterstützt. Wenn das Installationsprogramm erkennt, dass ein VDA unter Windows XP oder Windows Vista ausgeführt wird, wird ein anderes Installationsprogramm gestartet, mit dem die aktuelle VDA-Version für Windows XP oder Windows Vista (Version 5.6 FP1 mit einigen Hotfixes) installiert wird. Obwohl diese Maschinen und ihre Maschinenkataloge und Bereitstellungsgruppen nicht alle XenDesktop 7.x-Features verwenden können, können sie dennoch in der XenDesktop 7.x-Site ausgeführt werden.

VDAs, die unter Windows XP oder Windows Vista ausgeführt werden, unterstützen nicht die folgenden 7.x-Features:

- Konfiguration von App-V-Anwendungen von Studio
- Konfiguration von Receiver-Storefront-Adressen von Studio
- Automatische Unterstützung der Microsoft Windows KMS-Lizenzierung mit Maschinenerstellungsdiensten Die KMS-Lizenzierung wird mit den in <http://support.citrix.com/article/CTX128580> beschriebenen Schritten unterstützt.
- Vollständige Informationen in Director:
 - Anmeldezeiten und Anmeldeereignisse, die sich auf die Anmeldungsdauer auswirken, in den Ansichten "Dashboard", "Trends" und "Benutzerdetails".
 - Aufschlüsselung der Anmeldungsdauer für HDX-Verbindungsdauer, Authentifizierungsdauer, Profilladedauer, Dauer für das Laden des Gruppenrichtlinienobjekts, Anmeldeskriptdauer und Dauer für das Herstellen einer interaktiven Sitzung.
 - Einige Kategorien der Fehlerzustände für Maschinen und Verbindungen.
 - Aktivitäts-Manager in der Ansicht "Helpdesk" und "Benutzerdetails".

Sie können die folgenden Controllerversionen aktualisieren:

- 5.0
- 5.0 Service Pack 1
- 5,5
- 5.6
- 5.6 Feature Pack 1

Sie können die folgenden Director-Versionen aktualisieren:

- 1.0
- 1,1
- 2.0
- 2.1

Nachdem Sie ein manuelles Backup der Sitedatenbank erstellt haben, wie unter [How to backup and restore your XenDesktop Database](#) beschrieben, führen Sie mit einem aktualisierten Delivery Controller ein Upgrade der Datenbank

durch. Hierbei wird das Schema aktualisiert und die Daten werden migriert. Studio führt auch weitere Datenmigrationsschritte für die Dienste aus.

Das Installationsprogramm aktualisiert auch die folgenden Komponenten:

- Personal vDisk (PVD)
- Receiver
 - Wenn Receiver für Windows (Receiver.exe) auf einer Maschine installiert ist, wird das Programm auf Receiver für Windows 4.0 aktualisiert.
 - Wenn Receiver für Windows Enterprise (CitrixReceiverEnterprise.exe) auf einer Maschine installiert ist, wird das Programm auf Receiver für Windows Enterprise 3.4 aktualisiert

Komponenten, die ein separates Upgrade benötigen

Sie müssen die folgenden Komponenten außerhalb des integrierten Upgradeprozesses aktualisieren:

- Provisioning Services (PVS)
 - Aktualisieren Sie den Provisioning Services-Server mit dem rollenden Provisioning Services-Serverupgrade
 - Aktualisieren Sie den Provisioning Services-Client mit der Provisioning Services vDisk-Versionsverwaltung

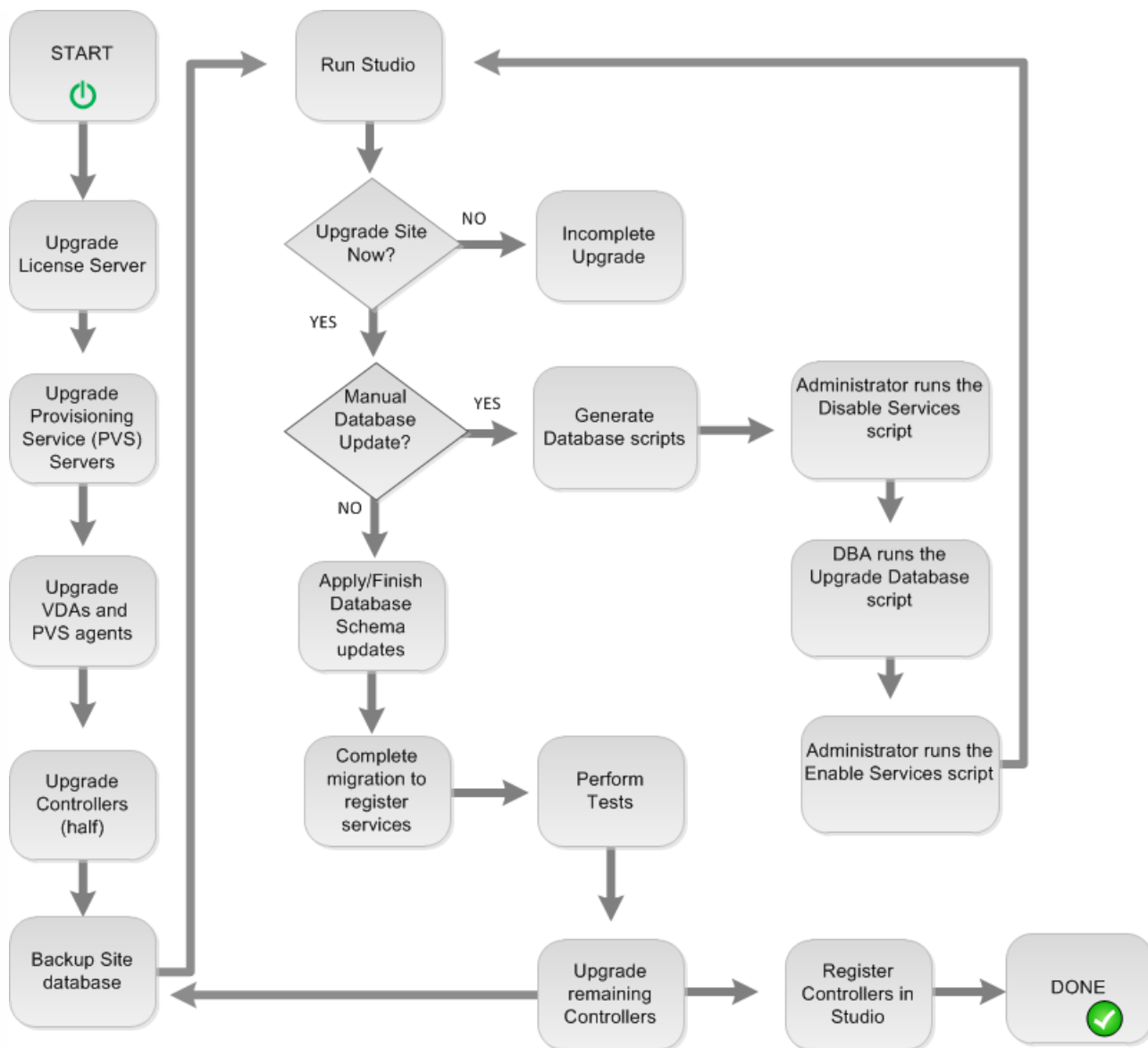
Wichtig: Wenn Sie XenDesktop 7.x- und XenDesktop 5.6-Sites gleichzeitig ausführen, führen Sie kein Upgrade auf Provisioning Services 7 durch.
- Microsoft System Center Virtual Machine Manager: XenDesktop 7.x unterstützt SCVMM 2012 und SCVMM 2012 SP1. XenDesktop 5.x unterstützt SCVMM 2008 R2 SP1. Führen Sie das Upgrade in der folgenden Reihenfolge durch, sodass XenDesktop ohne Ausfallzeit funktioniert.
 - Alle XenDesktop-Controller auf XenDesktop 5.6 FP1
 - SCVMM-Server auf SCVMM 2012
 - XenDesktop auf XenDesktop 7
 - Upgrade des SCVMM-Servers von 2012 auf 2012 SP1 (optional)
- Konfigurieren Sie für externe Webinterface-Server StoreFront zum Bereitstellen der Desktops, die vorher vom Webinterface bereitgestellt wurden. Weitere Informationen finden Sie in der StoreFront-Dokumentation. Fügen Sie XenDesktop nach dem Upgrade auf XenDesktop 7.x der StoreFront-Bereitstellung hinzu.

Nach dem Upgrade aller Komponenten können Sie optional Maschinenkataloge und Bereitstellungsgruppen mit Studio aktualisieren.

- Vorhandene Sites
 - Sie müssen das direkte Upgradeverfahren verwenden. Sie können Daten nicht von einer XenDesktop 5-Site in eine XenDesktop 7.x-Site importieren oder migrieren.
 - Parallele Sites (z. B. XenDesktop 5 und XenDesktop 7.1) sind zwar zulässig, Sie können jedoch eine XenDesktop 5-Site nicht mit XenDesktop Studio 7.x verwalten. Außerdem können Sie XenDesktop Studio 7.x nicht zusammen mit XenDesktop Studio 5 auf einem Computer installieren, es sei denn, Sie planen, die XenDesktop 5-Site zu aktualisieren. Sie können also verschiedene Studio-Installationen nicht parallel ausführen.
 - Aktualisieren Sie eine eigenständige Desktop Studio-Installation nicht auf Desktop Studio 7.x, wenn Sie noch nicht bereit sind, XenDesktop 7.x zu verwenden.
 - Sie können für Desktop Studio Express Edition kein Upgrade durchführen. Sie müssen vor dem Upgrade eine Lizenz für VDI, Enterprise oder Platinum Edition, erwerben und installieren.

- Sie können die Verwendung Ihrer XenApp Platinum-Lizenzen für Ihre Site nicht in Studio konfigurieren. Weitere Informationen über die Handhabung von XenApp Platinum-Lizenzen mit XenDesktop finden Sie unter [Verwalten der Lizenzierung](#).
- Delivery Controller: Sites mit einem Controller sind während des Upgrades nicht funktionsfähig. Sites mit mehr als einem Controller können während des Upgrades weiter ausgeführt werden. Durch das Upgrade wird nur für das Einrichten neuer Clientverbindungen während der letzten Datenbankaktualisierungsschritte eine kurze Unterbrechung verursacht. Zeitweise wird die Site möglicherweise mit reduzierter Kapazität ausgeführt, da weniger Controller verfügbar sind.
- VDAs: Sie können keinen neuen VDA auf einer Maschine mit Windows XP oder Windows Vista installieren. Sie müssen diese VDAs auf Version 5.6 Feature Pack 1 aktualisieren.
- Provisioning Services (PVS)
 - Bevor Sie ein Upgrade für eine Site ausführen, aktualisieren Sie alle Provisioning Services-Server, die auf Verwaltungsmaschinen in der Umgebung installiert sind. Weitere Informationen finden Sie unter [Aktualisieren von Provisioningservern](#).
 - Provisioning Services 7.x unterstützt nicht das Erstellen von neuen Desktops mit Versionen von XenDesktop 5. Obwohl die vorhandenen Desktops weiterhin ausgeführt werden, können Sie daher mit Provisioning Services keine neuen Desktops erstellen, bis das Upgrade der Controller und der Datenbank auf XenDesktop 7.x abgeschlossen ist. Wenn Sie ein Upgrade von Provisioning Services planen, lassen Sie diese Einschränkung bei der Planung des gesamten Site-Upgrades nicht außer Acht.
 - Wenn Sie XenApp Platinum-Lizenzen haben, können Sie Provisioning Services übertragen und mit einer XenDesktop 7.x App Edition-Lizenz verwenden. Dazu müssen Sie die richtigen Lizenzeinstellungen mit dem PowerShell SDK-Cmdlet auf dem Delivery Controller der Site konfigurieren, wie unter [Verwalten der Lizenzierung](#) erläutert.

Die folgenden Abbildungen zeigen die allgemeinen Schritte für das Upgrade von XenDesktop 5.



1. Aktualisieren Sie den Lizenzserver und die zugeordneten Lizenzdateien auf XenDesktop 7.x.
 - Wenn es einen separaten Lizenzserver gibt, aktualisieren Sie den Server zuerst.
 - Wenn sich der Lizenzserver auf einem Controller befindet, wird er zusammen mit den anderen Diensten aktualisiert.
2. Erstellen Sie ein Backup der Controller-Datenbank wie unter [How to Backup and Restore your XenDesktop Database](#) beschrieben.
3. Das Sichern von Vorlagen und das Upgrade von Hypervisor sind optional.
4. Aktualisieren Sie die Provisioning Services-Server und Agents.
Hinweis: Wenn Sie in einer XenDesktop 5.6-Site ein Upgrade auf Provisioning Services 7 durchgeführt haben, können Sie mit Provisioning Services keine neuen Desktops erstellen (vorhandene Desktops funktionieren jedoch weiterhin). Sie müssen die Controller und die Datenbank auf XenDesktop 7.x aktualisieren, um Provisioning Services zum Hinzufügen neuer Desktops verwenden zu können.
5. Virtual Desktop Agents (XenDesktop 5.x-VDAs), siehe Empfehlungen unter [XenDesktop 5-Upgradekomponenten](#). Sie können neue Software auf VDAs vor oder nach den Delivery Controllern installieren. Citrix empfiehlt jedoch, dass Sie dies, falls möglich, vor dem Upgrade der VDAs tun. Auf diese Weise können Sie schnell neue Features aktivieren, wenn die

Aktualisierung abgeschlossen ist.

6. Aktualisieren Sie die Hälfte (oder einige) der Controller. Bei Auswahl dieser Option werden auch andere Kernkomponenten aktualisiert.
 - Diese Controller sind dann für die vorhandene XenDesktop 5.x-Site unbrauchbar und können keine Maschinen mehr registrieren. Maschinen, die bei diesen Delivery Controllern registriert waren, werden bei den verfügbaren Delivery Controllern registriert.
 - Das Installationsprogramm überprüft, ob der Lizenzserver aktualisiert wurde und gibt eine Warnung aus, wenn der Lizenzserver nicht aktualisiert wurde.
7. Aktualisieren Sie eine Verwaltungsmaschine mit Studio oder verwenden Sie Studio auf einem der aktualisierten Controller.
8. Aktualisieren Sie die Datenbank mit Studio.

Wichtig: Citrix empfiehlt dringend, dass Sie manuell eine Sicherungskopie der Site erstellen, damit Sie die Site bei Problemen wiederherstellen können.

Aktualisieren Sie die alte Datenbank mit Studio 7.1 (Sie müssen Studio auf einem der aktualisierten Controller verwenden, nicht auf einem XenDesktop 5.x-Controller). Studio fordert Sie auf, eine Sicherungskopie zu erstellen zu prüfen, ob der Lizenzserver aktualisiert wurde.

Wenn Sie nicht über Administratorrechte für die Datenbank verfügen, wählen Sie Manuelles Upgrade aus, damit Sie die separaten manuellen Datenbankskripts für privilegierte Datenbankvorgänge verwenden können. Ausführliche Anweisungen finden Sie unter [XenDesktop 5-Upgradekomponenten](#).

Während die Dienste deaktiviert sind, können Controller keine neuen Verbindungen für die Site vermitteln. Halten Sie die Ausfallzeit daher so kurz wie möglich. Die Delivery Controller werden nach dem Abschluss dieses Schritts wieder aktiviert. Die Dienste müssen beim Delivery Controller registriert werden.

Nach dem Aktualisieren des Datenbankschemas führt Studio abschließende Datenmigrationsschritte für die Delivery Controller-Dienste aus.

Wenn der Prozess abgeschlossen ist:

- Studio führt Umgebungs- und Konfigurationstests aus und generiert einen HTML-Bericht für das Upgrade. Wenn diese Tests fehlschlagen, können Sie die Datenbank von der Sicherungskopie wieder herstellen und die ursprüngliche Datenbank verwenden. Nachdem Sie die Ursache der Probleme gefunden und die Probleme gelöst haben, führen Sie das Upgrade erneut aus.
 - Die aktualisierten Delivery Controller übernehmen nun die Registrierung der Maschinen.
9. Aktualisieren Sie die verbleibenden Delivery Controller (und Director) und die Verwaltungsmaschinen, um das Upgrade abzuschließen.
 10. Registrieren Sie die verbleibenden Controller wie unter
— *Aktualisieren der verbleibenden Delivery Controller*
beschrieben. Nach dem Abschluss des Upgrades aktualisieren Sie die Maschinenkataloge und die Bereitstellungsgruppen.

Nachdem das Upgrade und die Migration abgeschlossen sind, können Sie Umgebungs- und Konfigurationstests ausführen, um sicherzustellen, dass die Site einwandfrei funktioniert.

Zum Testen der aktualisierten Site klicken Sie im Sitekonfigurationsbereich auf der Studio-Seite "Häufige Aufgaben" auf Site testen.

Aktualisieren von Komponenten

Wenn Sie das Installationsprogramm ausführen (Automatische Auswahl), prüft der Assistent, ob bestimmte Sitekomponenten (z. B. Delivery Controller und VDAs) aktualisiert werden müssen. Wenn Sie an diesem Punkt einige Komponenten nicht aktualisieren, wird beim Ausführen von Studio gemeldet, welche Komponenten aktualisiert werden müssen. Sie können mit dem Verwalten der Site nicht fortfahren, bis Sie das Upgrade für diese Komponenten ausgeführt haben.

Abhängig von Ihrer Site können die auszuführenden Schritte und die Reihenfolge, in der sie ausgeführt werden müssen, variieren.

Wichtig: Bevor Sie mit dem Upgrade beginnen, erstellen Sie ein Backup Ihrer Datenbanken wie unter [How to Backup and Restore your XenDesktop Database](#) beschrieben.

Aktualisieren des Lizenzservers

Stellen Sie vor dem Upgrade des Lizenzservers sicher, dass das Subscription Advantage-Datum für Lizenzdateien kompatibel mit einer unterstützten Edition des Produkts ist.

1. Melden Sie sich beim Server mit einem lokalen Administratorkonto an und führen Sie das Installationsprogramm aus, indem Sie die Medien einlegen oder das ISO-Laufwerk für dieses Release bereitstellen und auf Automatische Auswahl doppelklicken.
2. Klicken Sie auf der Seite Willkommen auf Start. Der Assistent ermittelt, welche Komponenten aktualisiert werden müssen und zeigt die Seite Upgradeoptionen an, auf der die zu aktualisierenden Komponenten aktiviert werden.
3. Akzeptieren Sie die Lizenzvereinbarung.
4. Wenn der Assistent einen nicht kompatiblen Lizenzserver oder nicht kompatible Lizenzdateien erkennt, werden Sie aufgefordert, diese Lizenzkomponente zu aktualisieren. Aktualisieren Sie die Komponente und führen Sie dann das Installationsprogramm erneut aus, wie unter

— *Aktualisieren der Kernkomponenten*
erläutert.

Aktualisieren von Virtual Delivery Agent

Sie können Virtual Delivery Agents (VDAs) für Windows-Desktopbetriebssysteme nur aktualisieren, wenn Sie ein Upgrade von XenDesktop 5.x auf XenDesktop 7.x durchführen.

Für Remote-PC-Zugriff-Bereitstellungen empfiehlt Citrix, dass Sie für das Upgrade des VDAs auf dem PC die Befehlszeilenschnittstelle verwenden.

Wichtig: Für ein Upgrade eines VDAs der Version 5.x empfiehlt Citrix das Installieren des neuen VDAs auf einem sauberen Maschinenimage (ggf. auch auf einem älteren Golden Image, das keinen VDA enthält). Wenn dies nicht möglich ist, deinstallieren Sie den 5.x-VDA mit dem Windows-Feature zum Entfernen und Ändern von Programmen und installieren Sie dann die neue VDA-Version.

Befolgen Sie beim Installieren der neuen VDA-Version die Anweisungen in der Installationsdokumentation.

Aktualisieren von Kernkomponenten

Das Installationsprogramm aktualisiert automatisch Kernkomponenten wie Delivery Controller, Studio und Director, wenn sie bereits installiert sind.

1. Melden Sie sich beim Server mit einem lokalen Administratorkonto an und führen Sie das Installationsprogramm aus, indem Sie die Medien einlegen oder das ISO-Laufwerk für dieses Release bereitstellen und auf Automatische Auswahl doppelklicken.
2. Klicken Sie auf der Seite Willkommen auf Start. Der Assistent ermittelt, welche Komponenten aktualisiert werden müssen und zeigt die Seite Upgradeoptionen an, auf der die zu aktualisierenden Komponenten aktiviert werden.

3. Klicken Sie auf Delivery Controller.
Hinweis: Wenn die XenDesktop Express Edition erkannt wird, werden Sie aufgefordert, eine Lizenz für eine unterstützte Version zu beziehen und zu installieren. Die Aktualisierung kann erst fortgesetzt werden, wenn eine Lizenz für VDI, für die Enterprise- oder Platinum-Edition installiert wurde.
4. Akzeptieren Sie die Lizenzvereinbarung.
5. Überprüfen Sie die Upgradeschritte, klicken Sie auf Ich bin zum Fortfahren bereit, und klicken Sie dann auf Weiter.
6. Prüfen Sie auf der Seite Kernkomponenten, welche Komponenten aktualisiert werden müssen.
7. Überprüfen Sie auf der Seite Firewall die Standardports und konfigurieren Sie Firewallregeln.
8. Überprüfen Sie auf der Seite Upgrade die zu installierenden Voraussetzungen und die zu aktualisierenden Komponenten und klicken Sie dann auf Upgrade.
9. Auf der Seite Upgrade fertig stellen wird nach dem Upgrade eine der folgenden Meldungen angezeigt:
 - Erfolg: **Upgrade erfolgreich** wird angezeigt, wenn das Upgrade ohne Fehler abgeschlossen wurde.
 - Fehler: **Upgrade fehlgeschlagen** wird mit einer Liste der fehlerhaften Komponenten angezeigt. Klicken Sie auf Warum ist dieser Schritt fehlgeschlagen?, um zu sehen, wie Sie das Problem beheben können. Komponenten, die erfolgreich installiert wurden, bleiben gespeichert. Sie müssen nicht neu installiert werden.
10. Wählen Sie Studio starten aus, um Studio nach dem Abschluss des Upgrades zu starten, und klicken Sie dann auf Fertig stellen.

Aktualisieren der Datenbank mit Studio

Nach der Aktualisierung der Kernkomponenten aktualisieren Sie die Datenbank mit Studio.

Manuelles Aktualisieren der Datenbank

Damit die Ausfallzeit der Site bei einem manuellen Upgrade so kurz wie möglich ist, müssen Studio-Administrator und Datenbankadministrator eng zusammenarbeiten. Für diesen Vorgang müssen Sie ein Skript ausführen, das die Dienste vorübergehend deaktiviert, während die Skripts für das manuelle Upgrade ausgeführt werden. Idealerweise sollte der Studio-Administrator unmittelbar nach dem Ausführen der Skripts die Dienste aktivieren und das Upgrade mit Studio abschließen.

Ein manuelles Upgrade erfordert Folgendes:

- Kompatiblen Lizenzserver und Lizenzdateien
- Sicherungskopie der Datenbank
- Das Ausführen der generierten Skripts in der folgenden Reihenfolge:
 - DisableServices.ps1: PowerShell-Skript, das vom Studio-Administrator auf einem Controller ausgeführt wird
 - UpgradeDatabase.sql: SQL-Skript, das vom Datenbankadministrator mit einem bevorzugten Tool ausgeführt werden muss (z. B. SQL Server Management Studio)
 - EnableServices.ps1: PowerShell-Skript, das vom Studio-Administrator auf einem Controller ausgeführt wird

Sie aktualisieren die Datenbank manuell wie folgt:

1. Starten Sie Studio. Der Assistent ermittelt, welche Komponenten aktualisiert werden müssen und zeigt die Seite Obligatorisches Upgrade an.
2. Wählen Sie Manuelles Siteupgrade durchführen aus.
3. Der Assistent überprüft die Kompatibilität des Lizenzservers. Stellen Sie sicher, dass der Lizenzserver und die Lizenzdateien kompatibel mit Ihrer XenDesktop 7.x-Version sind. Aktivieren Sie das Bestätigungskontrollkästchen und klicken Sie auf Weiter.
4. Der Assistent fordert Sie zur Sicherung der Datenbank auf. Wenn Sie die Sicherung ausgeführt haben, aktivieren Sie das Bestätigungskontrollkästchen und klicken Sie auf Weiter.
 - Der Assistent erstellt die manuellen Upgradeskripts, die ausgeführt werden müssen, und zeigt sie in einem Fenster an.

- Auf der Seite "Obligatorisches Upgrade" wird eine Prüfliste der manuellen Upgradeschritte angezeigt.
5. Stellen Sie sicher, dass Sie die Aufgaben auf der Prüfliste abgeschlossen haben und klicken Sie auf Schließen Sie das Upgrade ab und kehren Sie zu 'Häufige Aufgaben' zurück.

Aktualisieren der restlichen Delivery Controller

Aktualisieren Sie alle zusätzlichen Controller in der Site.

1. Klicken Sie auf der Seite Häufige Aufgaben auf Verbleibende Controller aktualisieren. Eine Liste der Controller wird angezeigt.
2. Führen Sie alle zuvor beschriebenen Aufgaben, angefangen mit
— *Aktualisieren der Kernkomponenten*
, auf jedem Controller aus.
3. Wenn Sie alle Controller aktualisiert haben, klicken Sie auf Ich habe die verbleibenden Delivery Controller aktualisiert und klicken Sie auf Fertig stellen.
4. Schließen Sie Studio und öffnen Sie Studio erneut, damit die Änderungen implementiert werden.
5. Wählen Sie auf der Seite Häufige Aufgaben im Abschnitt "Sitekonfiguration" die Option Registrierung durchführen aus. Das Registrieren der verbleibenden Controller macht die Controller und die Dienste auf der Site verfügbar.

Upgrade von XenDesktop 7

Sep 29, 2015

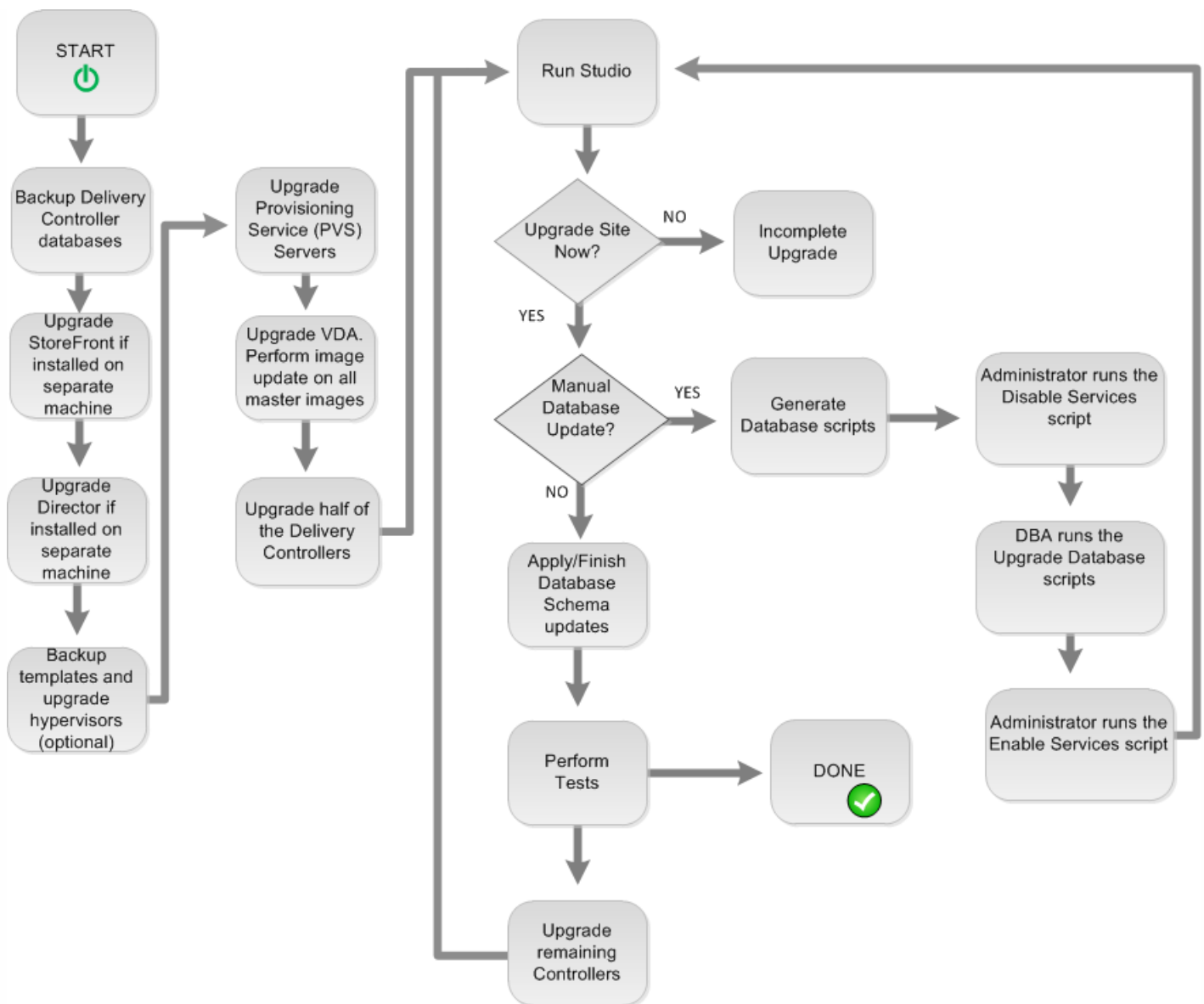
Prüfen Sie die folgenden Informationen, bevor Sie ein Upgrade auf XenDesktop 7 durchführen:

- Vorhandene Sites: Sie müssen das als "direktes Upgrade" bezeichnete Verfahren zum Aktualisieren auf XenDesktop 7.1 oder 7.5 verwenden.
- VDAs: Virtual Desktop Agents, die unter Windows XP oder Windows Vista ausgeführt werden, können nicht auf Virtual Delivery Agents Version 7.1 aktualisiert werden. Sie müssen diese VDAs auf die vom Installer bereitgestellte Windows XP- oder Windows Vista-Version aktualisieren oder ein Upgrade auf Version 5.6 Feature Pack 1 durchführen.
- Microsoft System Center Virtual Machine Manager (VMM): XenDesktop 7.1 und 7.5 unterstützen VMM 2012, VMM 2012 SP1 und VMM 2012 R2.

Wenn Sie das Installationsprogramm ausführen (Automatische Auswahl), prüft der Assistent, ob bestimmte Sitekomponenten (z. B. die Delivery Controller, Director und VDAs) aktualisiert werden müssen. Wenn Sie an diesem Punkt einige Komponenten nicht aktualisieren, wird beim Ausführen von Studio eine Komponentenprüfung durchgeführt und Sie werden benachrichtigt, wenn Komponenten aktualisiert werden müssen. Sie können mit dem Verwalten der Site nicht fortfahren, bis Sie das Upgrade für diese Komponenten ausgeführt haben.

Wichtig: Bevor Sie mit dem Upgrade beginnen, erstellen Sie ein Backup Ihrer Datenbanken wie unter [How to backup and restore your XenDesktop Database](#) beschrieben.

Die folgende Abbildung zeigt eine Übersicht über die erforderlichen Prozesse zum Aktualisieren einer XenDesktop 7-Site auf XenDesktop 7.x, wenn deren Studio-Komponenten auf unterschiedlichen Maschinen bereitgestellt wurden.



Führen Sie die Upgradeschritte in der folgenden Reihenfolge aus.

Stellen Sie sicher, dass das Subscription Advantage-Datum für Lizenzen nicht vor 2013.0522 liegt.

Wenn StoreFront auf einer separaten Maschine bereitgestellt wird, führen Sie die unter [Aktualisieren von StoreFront](#) beschriebenen Schritte aus.

Wenn Director auf einer separaten Maschine bereitgestellt wird, führen Sie die unter [Aktualisieren der Kernkomponenten](#) erläuterten Schritte aus.

Wenn Sie Provisioning Services verwenden, führen Sie die unter [Aktualisieren von Provisioning Services](#) erläuterten Schritte

aus.

Ein Upgrade von auf einer physischen Maschine (einschließlich Remote-PC-Zugriff) installierten VDAs der Version 7 oder 7.1 auf Version 7.5 müssen Sie über die Befehlszeile mit dem folgenden Parameter starten:

```
/EXCLUDE "Personal vDisk","Machine Identity Service"
```

Das nachfolgende Beispiel zeigt das manuelle Upgrade eines physischen VDAs.

1. Wenn Sie ein physisches Installationsmedium verwenden, legen Sie es in das DVD-Laufwerk ein. Wenn Sie eine Netzwerkfreigabe verwenden, binden Sie das Medium als Netzlaufwerk ein. In diesem Beispiel wird davon ausgegangen, dass das Installationsmedium in Laufwerk D ist.
2. Öffnen Sie ein Eingabeaufforderungsfenster.
3. Geben Sie Folgendes ein:
`D:\x64\XenDesktop Setup\XenDesktopVdaSetup.exe /EXCLUDE "Personal vDisk","Machine Identity Service"`
Das VDA-Upgrade wird damit gestartet und ausgeführt wie unter
— Aktualisieren von Virtual Delivery Agent
beschrieben.

Für Remote-PC-Zugriff-Bereitstellungen empfiehlt Citrix, dass Sie einen VDA auf einem Remote-PC über die Befehlszeile aktualisieren. Weitere Informationen finden Sie unter [Installieren an der Befehlszeile](#).

1. Melden Sie sich beim Server mit einem lokalen Administratorkonto an und führen Sie das Installationsprogramm aus, indem Sie die Medien einlegen oder das ISO-Laufwerk für dieses Release bereitstellen und auf Automatische Auswahl doppelklicken.
2. Klicken Sie auf der Seite Willkommen auf Start. Der Assistent ermittelt, welche Komponenten aktualisiert werden müssen und zeigt die Seite Upgradeoptionen an, auf der die zu aktualisierenden Komponenten aktiviert werden.
3. Wählen Sie auf der Seite Upgradeoptionen eine der folgenden Optionen aus:
 - Virtual Delivery Agent für Windows-Desktopbetriebssysteme für Desktopbetriebssysteme und frühere XenDesktop-Versionen
 - Virtual Delivery Agent für Windows-Serverbetriebssysteme für Serverbetriebssysteme
4. Überprüfen Sie auf der Seite Firewall die Standardports und konfigurieren Sie Firewallregeln.
5. Überprüfen Sie auf der Seite Zusammenfassung die zu installierenden Voraussetzungen sowie die zu aktualisierenden Komponenten, und klicken Sie dann auf Aktualisieren.
6. Auf der Seite Upgrade fertig stellen wird nach dem Upgrade eine der folgenden Meldungen angezeigt:
 - Erfolg: **Upgrade erfolgreich** wird angezeigt, wenn das Upgrade ohne Fehler abgeschlossen wurde.
 - Fehler: **Upgrade fehlgeschlagen** wird mit einer Liste der fehlerhaften Komponenten angezeigt. Klicken Sie auf Warum ist dieser Schritt fehlgeschlagen?, um zu sehen, wie Sie das Problem beheben können. Komponenten, die erfolgreich installiert wurden, bleiben gespeichert. Sie müssen nicht neu installiert werden.
7. Klicken Sie auf Fertigstellen, um das Upgrade abzuschließen.

Citrix empfiehlt, dass Sie die Delivery Controller wie folgt aktualisieren:

1. Aktualisieren Sie die Hälfte der Delivery Controller der Site.
2. Verwenden Sie die aktualisierte Version von Studio für das Siteupgrade, wie unter
— Aktualisieren der Kernkomponenten

erläutert.

3. Aktualisieren Sie die restlichen Delivery Controller.

Verwenden Sie Studio zum Aktualisieren der Datenbank.

Automatisches Aktualisieren von Datenbanken

1. Starten Sie Studio. Der Assistent ermittelt, welche Komponenten aktualisiert werden müssen und zeigt die Seite Obligatorisches Upgrade an.
2. Wählen Sie Siteupgrade automatisch starten aus.
3. Geben Sie an der Eingabeaufforderung Bereit zum Upgrade ein.
Der Assistent zeigt eine Fortschrittsanzeige für das Upgrade an. Nachdem das Upgrade abgeschlossen ist, führt der Assistent Tests aus. Dies dauert einige Minuten.
4. Im Fenster Siteupgrade abgeschlossen können Sie sich einen Datenmigrationsbericht anzeigen lassen. Klicken Sie dann auf Fertig stellen.
5. Klicken Sie auf der Seite Upgrade erfolgreich auf Upgrade fertig stellen und zum Siteüberblick zurückkehren.

Manuelles Aktualisieren von Datenbanken

Damit die Ausfallzeit der Site bei einem manuellen Upgrade so gering wie möglich ist, ist es wichtig, dass der XenDesktop-Administrator eng mit dem Datenbankadministrator zusammenarbeitet und sich abspricht. Dieser Prozess erfordert die Ausführung eines Skripts, das die XenDesktop-Dienste vorübergehend deaktiviert, während die manuellen Upgradeskripts vom Datenbankadministrator mit einem bevorzugten Tool ausgeführt werden (z. B. SQL Server Management Studio). Idealerweise sollte der XenDesktop-Administrator unmittelbar nach dem Ausführen der Skripts die XenDesktop-Dienste aktivieren und das Upgrade mit Studio abschließen.

Das manuelle Upgrade erfordert Folgendes:

- Sichern der Datenbanken
 - Das Ausführen der generierten Skripts in der folgenden Reihenfolge:
1. DisableServices.ps1: PowerShell-Skript, das vom XenDesktop-Administrator auf einem XenDesktop-Controller ausgeführt werden muss.
 2. UpgradeSiteDatabase.sql: SQL-Skript, das auf dem Speicherort der Sitedatenbanken ausgeführt wird.
 3. UpgradeMonitorDatabase.sql: SQL-Skript, das auf dem Speicherort der Überwachungsdatenbanken ausgeführt wird.
 4. UpgradeLoggingDatabase.sql: SQL-Skript, das auf dem Speicherort der Protokollierungsdatenbank ausgeführt wird.
Hinweis: Führen Sie dieses Skript nur aus, wenn Änderungen an der Protokollierungsdatenbank vorgenommen werden.
Beispiel: Führen Sie das Skript nach dem Anwenden eines Hotfixes aus.
 5. EnableServices.ps1: PowerShell-Skript, das vom XenDesktop-Administrator auf einem XenDesktop-Controller ausgeführt werden muss.

Durchführen eines manuellen Upgrades

1. Starten Sie Studio. Der Assistent ermittelt, welche Komponenten aktualisiert werden müssen und zeigt die Seite Obligatorisches Upgrade an.
2. Wählen Sie Manuelles Siteupgrade durchführen aus.
3. Der Assistent überprüft die Kompatibilität des Lizenzservers. Stellen Sie sicher, dass der Lizenzserver und die

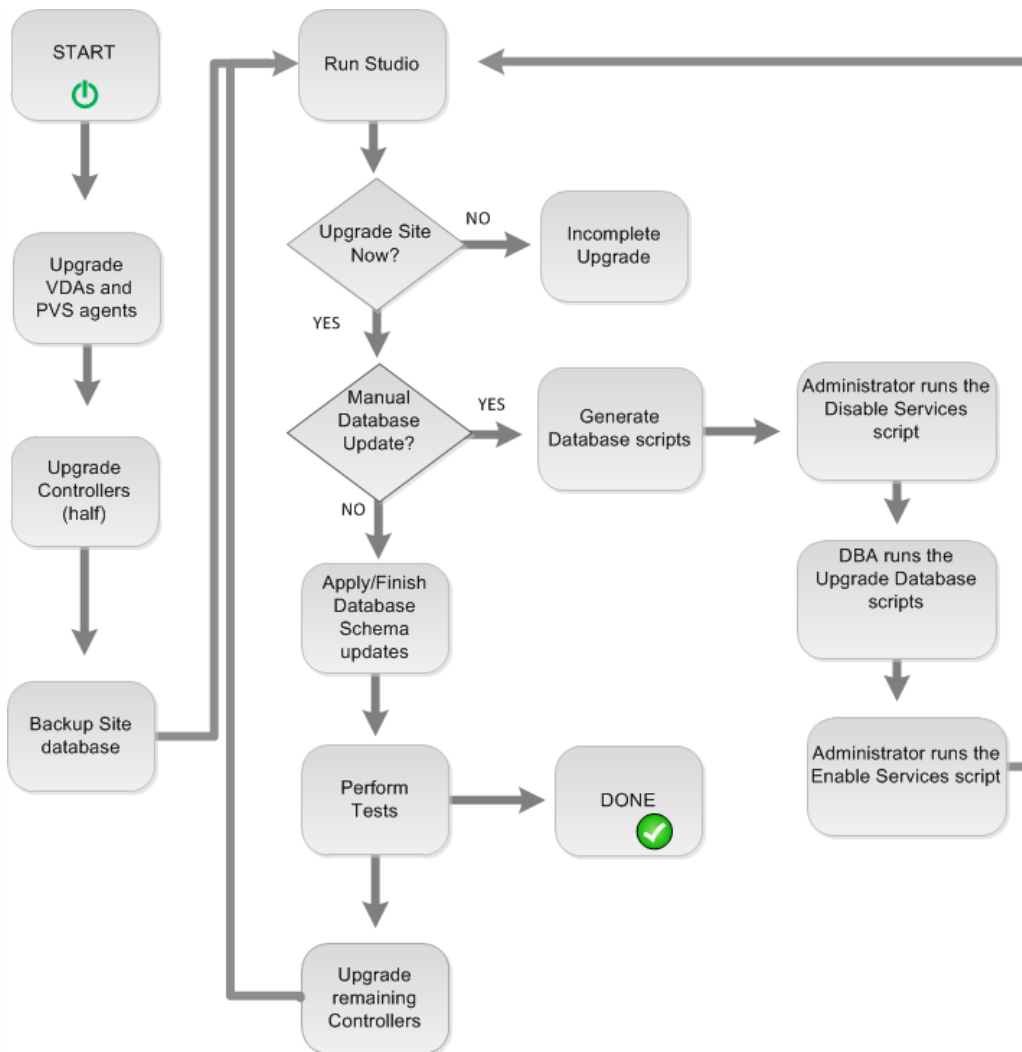
Lizenzdateien kompatibel mit Ihrer XenDesktop 7.x-Version sind. Aktivieren Sie das Bestätigungskontrollkästchen und klicken Sie auf Weiter.

4. Der Assistent fordert Sie zur Sicherung der Datenbank auf. Wenn Sie die Sicherung ausgeführt haben, aktivieren Sie das Bestätigungskontrollkästchen und klicken Sie auf Weiter.
 - Der Assistent erstellt die manuellen Upgradeskripts, die ausgeführt werden müssen, und zeigt sie in einem Fenster an.
 - Auf der Seite "Obligatorisches Upgrade" wird eine Prüfliste der manuellen Upgradeschritte angezeigt.
5. Stellen Sie sicher, dass Sie die Aufgaben auf der Prüfliste abgeschlossen haben und klicken Sie auf Schließen Sie das Upgrade ab und kehren Sie zu 'Häufige Aufgaben' zurück.

Das Installationsprogramm aktualisiert automatisch Kernkomponenten wie Delivery Controller, Studio und Director, wenn sie bereits installiert sind.

1. Melden Sie sich beim Server mit einem lokalen Administratorkonto an und führen Sie das Installationsprogramm aus, indem Sie die Medien einlegen oder das ISO-Laufwerk für dieses Release bereitstellen und auf Automatische Auswahl doppelklicken.
2. Klicken Sie auf der Seite Willkommen auf Start. Der Assistent ermittelt, welche Komponenten aktualisiert werden müssen und zeigt die Seite Upgradeoptionen an, auf der die zu aktualisierenden Komponenten aktiviert werden.
3. Klicken Sie auf Delivery Controller.
4. Akzeptieren Sie die Lizenzvereinbarung.
5. Überprüfen Sie die Upgradeschritte, klicken Sie auf Ich bin zum Fortfahren bereit, und klicken Sie dann auf Weiter.
6. Prüfen Sie auf der Seite Kernkomponenten, welche Komponenten aktualisiert werden müssen.
7. Überprüfen Sie auf der Seite Firewall die Standardports und konfigurieren Sie Firewallregeln.
8. Überprüfen Sie auf der Seite Upgrade die zu installierenden Voraussetzungen und die zu aktualisierenden Komponenten und klicken Sie dann auf Upgrade.
9. Auf der Seite Upgrade fertig stellen wird nach dem Upgrade eine der folgenden Meldungen angezeigt:
 - Erfolg: **Upgrade erfolgreich** wird angezeigt, wenn das Upgrade ohne Fehler abgeschlossen wurde.
 - Fehler: **Upgrade fehlgeschlagen** wird mit einer Liste der fehlerhaften Komponenten angezeigt. Klicken Sie auf Warum ist dieser Schritt fehlgeschlagen?, um zu sehen, wie Sie das Problem beheben können. Komponenten, die erfolgreich installiert wurden, bleiben gespeichert. Sie müssen nicht neu installiert werden.
10. Klicken Sie auf Fertigstellen, um das Upgrade abzuschließen.
11. Nach dem Abschluss des Upgrades von XenDesktop aktualisieren Sie die Maschinenkataloge wie unter [Aktualisieren eines Maschinenkatalogs](#) erläutert, und aktualisieren Sie die Bereitstellungsgruppen wie unter [Aktualisieren einer Bereitstellungsgruppe](#) erläutert.

Die folgende Abbildung zeigt eine Übersicht über die erforderlichen Prozesse zum Aktualisieren einer Site, deren Komponenten alle auf demselben Computer bereitgestellt wurden.



Führen Sie die Upgradeschritte in der folgenden Reihenfolge aus:

1. Stellen Sie sicher, dass das Subscription Advantage-Datum für Lizenzen nicht vor 2013.0522 liegt.
2. Erstellen Sie ein Backup der Controller-Datenbanken wie unter [How to backup and restore your XenDesktop Database](#) beschrieben.
3. Das Sichern von Vorlagen und das Upgrade von Hypervisor sind optional.
4. Aktualisieren Sie die PVS-Server und -Agents wie unter
— *Aktualisieren von Provisioning Services* erläutert.
5. Aktualisieren Sie die Kernkomponenten wie unter
— *Aktualisieren der Kernkomponenten* erläutert.
6. Aktualisieren Sie die VDAs wie unter
— *Aktualisieren von Virtual Delivery Agent* erläutert.
7. Aktualisieren Sie die Datenbank mit Studio wie unter
— *Aktualisieren der Datenbanken mit Studio* erläutert.
8. Aktualisieren Sie die restlichen Delivery Controller.
9. Nach dem Abschluss des Upgrades von XenDesktop aktualisieren Sie die Maschinenkataloge wie unter [Aktualisieren eines Maschinenkatalogs](#) erläutert, und aktualisieren Sie die Bereitstellungsgruppen wie unter [Aktualisieren einer Bereitstellungsgruppe](#) erläutert.

[Bereitstellungsgruppe](#) erläutert.

Nachdem das Upgrade und die Migration abgeschlossen sind, können Sie Umgebungs- und Konfigurationstests ausführen, um sicherzustellen, dass die Site einwandfrei funktioniert.

Zum Testen der aktualisierten Site klicken Sie im Sitekonfigurationsbereich auf der Studio-Seite "Häufige Aufgaben" auf Site testen.

Migration

Sep 29, 2015

Sie können die Daten und Einstellungen von einer XenDesktop 4-Farm in eine XenDesktop 7.x-Site mit dem Migrationstool übertragen. Dieses umfasst folgende Komponenten:

- Das Exporttool XdExport zum Export der XenDesktop 4-Farmdaten.
- Einen XML-Editor zum Überprüfen und Bearbeiten der XML-Datei, deren Standardname XdSettings.xml ist.
- Das Importtool XdImport zum Importieren der Daten durch Ausführen des PowerShell-Skripts Import-XdSettings.ps1.

Für eine erfolgreiche Verwendung des Migrationstools müssen beide Bereitstellungen in folgender Hinsicht übereinstimmen:

- Hypervisor Version (beispielsweise XenServer 6.2)
- Active Directory-Umgebung

Einschränkungen für XenApp

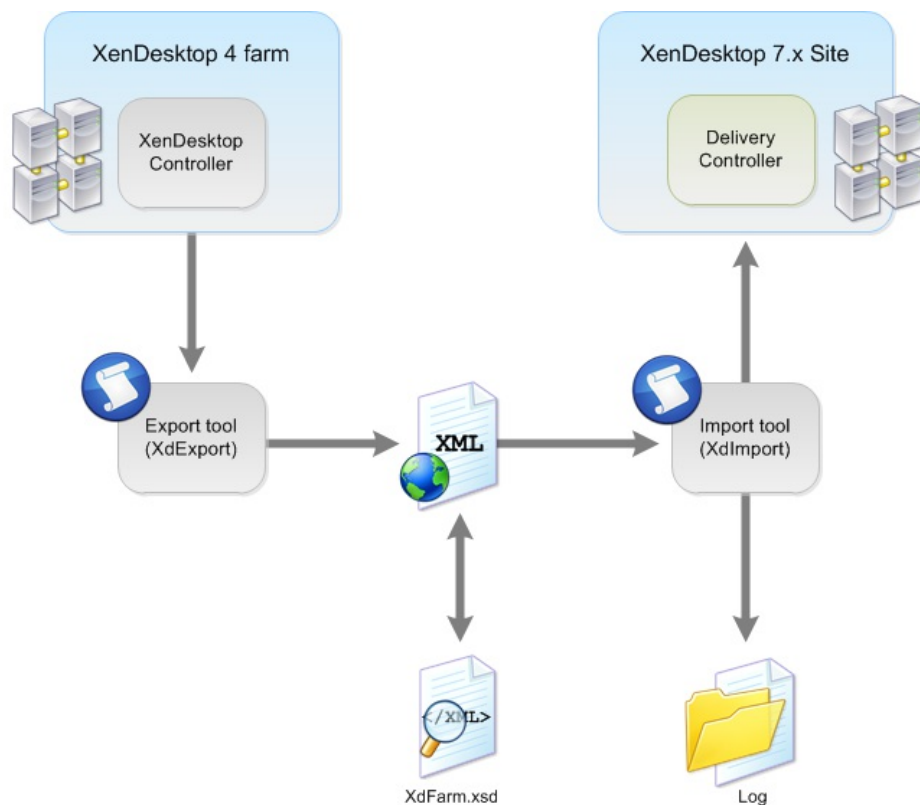
- Sie können keine früheren XenApp-Versionen zur aktuellen Version migrieren.
- Sie können keine XenApp-Farmen zu XenDesktop migrieren.
- Sie können XenDesktop 4 nicht zu XenApp migrieren.

Voraussetzungen

Führen Sie die folgenden Aufgaben vor der Migration durch:

- Informieren Sie sich darüber, welche Daten exportiert und importiert werden können und welche Konsequenzen dies für Ihre eigene Bereitstellung hat. Informationen zu den zu exportierenden und importierenden Datentypen finden Sie unter [Details zum Datenimport und Datenexport](#).
- Citrix empfiehlt, dass Sie manuell eine Sicherungskopie der Sitedatenbank erstellen, damit Sie sie bei Problemen wieder herstellen können.
- Richten Sie eine XenDesktop 7.x-Site einschließlich Datenbank ein. Vollständige Informationen zu den Datenbankanforderungen finden Sie unter [Datenbankanforderungen](#).
- Für die Migration von XenDesktop 4 auf XenDesktop 7.x müssen alle VDAs auf der XenDesktop 5.x-Ebene sein, damit sie mit XenDesktop 4- und XenDesktop 7.x-Controllern kompatibel sind. Wenn die Controllerinfrastruktur nur XenDesktop 7.x ausführt, können Windows 7-VDAs auf XenDesktop 7.x aktualisiert werden. Weitere Informationen finden Sie unter [Migrationsbeispiele](#).

In der folgenden Abbildung wird der Migrationsvorgang zusammengefasst.



1. Setzen Sie in der Desktop Studio-Konsole auf dem XenDesktop 4-Controller alle zu exportierenden Maschinen in den Wartungsmodus.
2. Exportieren Sie Daten und Einstellungen von der XenDesktop 4-Farm mit dem Exporttool in eine XML-Datei. Weitere Informationen finden Sie unter [Exportieren aus einer XenDesktop 4-Farm](#) aus.
3. Bearbeiten Sie die XML-Datei so, dass sie nur die Daten und Einstellungen enthält, die Sie in Ihre XenDesktop 7.x-Site importieren möchten. Weitere Informationen zum Bearbeiten der XML-Datei finden Sie unter [Bearbeiten der XML-Datei des Migrationstools](#).
4. Importieren Sie die Daten und Einstellungen aus der XML-Datei in Ihre XenDesktop 7.x-Site. Weitere Informationen zur Verwendung des Importtools für diesen Schritt finden Sie unter [Importieren von XenDesktop 4-Daten](#).
5. Wiederholen Sie die Schritte 4 und 5, um weitere Änderungen vorzunehmen. Nach dem Vornehmen von Änderungen können Sie zusätzliche Desktops in vorhandene Desktopgruppen importieren. Verwenden Sie hierfür die Option "Mergedesktops" wie unter [Importieren von XenDesktop 4-Daten](#) beschrieben.
6. Führen Sie die nach der Migration erforderlichen Schritte durch wie unter [Aufgaben nach der Migration](#) beschrieben.

Beispiele für die Migration finden Sie unter [Migrationsbeispiele](#).

Neue Konzepte und Features

Im Vergleich zu XenDesktop 4 enthält dieses Release zahlreiche neue Features und neue Konzepte. Weitere Informationen zu neuen Konzepten, Terminologie und Systemanforderungen finden Sie unter [Info über XenDesktop](#).

Beispiel 1: Migrieren einer großen XenDesktop 4-Farm auf eine XenDesktop 7-Site

In diesem Beispiel wird die gesamte XenDesktop 4-Farm auf XenDesktop 7 aktualisiert.

Ausgangsbedingungen

Derzeit wird eine XenDesktop 4-Farm verwendet. Die XenDesktop 4-Farm hat 50 Desktopgruppen, wobei jede Gruppe durchschnittlich 100 Desktops hat. Die XenDesktop 4-Desktops werden über Provisioning Services (PVS) bereitgestellt und die Maschinen werden auf VMware ESX-Hypervisors ausgeführt. Als VDA-Software ist die XenDesktop 4-VDA-Software auf allen VMs installiert.

Migrationsschritte

1. Aktualisieren Sie alle XenDesktop 4-VDAs auf die XenDesktop 5.6 Feature Pack 1-VDA-Software. Dann können die VDAs sowohl beim XenDesktop 4-Controller als auch beim XenDesktop 7-Delivery Controller registriert werden.
 - Weitere Informationen zu Windows 7-VDAs, die auf XenDesktop 4 ausgeführt werden, finden Sie unter [Durchführen eines Upgrades von Virtual Desktop Agent auf einer VM oder einem Blade-Computer](#).
 - Weitere Informationen zu VDAs unter Windows XP und Windows Vista, die auf XenDesktop 4 ausgeführt werden, finden Sie unter [Installieren einer früheren Version von Virtual Desktop Agent unter Windows XP oder Windows Vista](#).
2. Stellen Sie sicher, dass sich alle Benutzer von der vorhandenen XenDesktop 4-Farm abgemeldet haben.
3. Stellen Sie sicher, dass alle Maschinen in den Wartungsmodus gesetzt wurden.
4. Führen Sie das Exporttool (XdExport) auf der vorhandenen XenDesktop 4-Farm aus.
5. Installieren Sie XenDesktop 7.
 1. Führen Sie die XenDesktop 7-Sitekonfiguration mit Studio durch. Wählen Sie den Modus für eine vollständige Produktionssite im Bereitstellungsassistenten.
 2. Aktualisieren Sie den Provisioning Services-Server und die PVS-Agents, wenn PVS Teil der Bereitstellung ist.
 3. Aktualisieren Sie den Lizenzserver (und die zugehörigen Lizenzen).
6. Entzippen Sie das Importtool (XdImport) in einem lokalen Verzeichnis auf dem XenDesktop 7-Delivery-Controller.
7. Kopieren Sie die in Schritt 3 mit dem Exporttool erstellte XML-Datei (XdSettings.xml) in das lokale Verzeichnis.
8. Starten Sie im PowerShell-Bereich des Studio-Stammknotens der XenDesktop 7-Site eine PowerShell-Sitzung.
9. Führen Sie das Importtool (XdImport) aus, sodass die Anmeldeinformationen der zugeordneten Hypervisors und der Pfad der XML-Datei übergeben werden.
10. Erstellen Sie manuell Administratoreinstellungen über den Administratorknoten in Studio, wie unter [Administratoren](#) beschrieben.
11. Ändern Sie die importierten Desktops so, dass sie die registrierungsbasierte Controller-Discovery verwenden, und verweisen Sie sie auf den neuen XenDesktop 7-Delivery Controller.
12. Für VDAs, die unter Windows 7 ausgeführt werden, empfiehlt Citrix, dass Sie diese VDAs aktualisieren, damit sie den Desktopbetriebssystem-VDA von XenDesktop 7 verwenden. Dies bietet Zugriff auf alle XenDesktop 7 Features. Nach dem Upgrade der VDA-Software auf XenDesktop 7 müssen Sie für Maschinen in einem Katalog oder einer Bereitstellungsgruppe den Katalog und die Bereitstellungsgruppen aktualisieren, wie unter [Aktualisieren eines Maschinenkatalogs](#) und [Aktualisieren einer Bereitstellungsgruppe](#) beschrieben.
13. Beenden Sie den Wartungsmodus für Bereitstellungsgruppen.
14. Konfigurieren Sie StoreFront, um die zuvor vom Webinterface bereitgestellten Desktops zur Verfügung zu stellen. Weitere Informationen zum Installieren und Konfigurieren von StoreFront finden Sie unter [Installieren und Einrichten von StoreFront](#).

Beispiel 2: Schließen Sie einen XenDesktop 4-Farmexport mit einem teilweisen Import in eine XenDesktop 7.1-Site ab

Bei diesem Beispiel wird eine geläufigere Methode angewendet, bei der die Migration in mehreren Schritten erfolgt, wobei bei jedem Schritt ein Teil der zu migrierenden verbleibenden Desktops migriert wird.

Ausgangsbedingungen

Derzeit wird eine XenDesktop 4-Farm verwendet und eine XenDesktop 7.1-Site wurde bereits erstellt und wird ebenfalls verwendet. Die XenDesktop 4-Farm hat 50 Desktopgruppen, wobei jede Gruppe durchschnittlich 100 Desktops hat. Die XenDesktop 4-Desktops werden über Provisioning Services (PVS) bereitgestellt und die Maschinen werden auf Citrix XenServer-Hypervisors ausgeführt. Als VDA-Software wurde XenDesktop 4 auf allen virtuellen XenDesktop 4-Maschinen installiert.

Migrationsschritte

1. Führen Sie das Exporttool (XdExport) auf der vorhandenen XenDesktop 4-Farm aus.
 1. Entzippen Sie das Exporttool (XdExport) auf einer Desktop Delivery Controller-Maschine in der Farm.
 2. Führen Sie das Exporttool als Citrix Administrator ohne Parameter aus.
2. Kopieren und bearbeiten Sie die resultierende XML-Datei, sodass sie nur die Gruppen und Desktops enthält, die Sie migrieren möchten.
3. Stellen Sie sicher, dass alle Benutzer der zu migrierenden Desktops von der vorhandenen Farm abgemeldet sind.
4. Stellen Sie sicher, dass alle zu migrierenden Desktops in XenDesktop 4 im Wartungsmodus sind.
5. Entzippen Sie das Importtool (XdImport) in einem lokalen Verzeichnis auf dem XenDesktop 7.1-Delivery-Controller.
6. Kopieren Sie die bearbeitete XML-Datei in das lokale Verzeichnis.
7. Starten Sie im PowerShell-Bereich des Studio-Stammknotens der XenDesktop 7.1-Site eine PowerShell-Sitzung.
8. Führen Sie das Importtool (XdImport) aus, sodass die Anmeldeinformationen der zugeordneten Hypervisors und der Pfad der XML-Datei übergeben werden.
9. Erstellen Sie manuell Administratoreinstellungen über den Administratorknoten in Studio, wie unter [Administratoren](#) beschrieben.
10. Ändern Sie die importierten Desktops so, dass sie die registrierungsbasierte Controller-Discovery verwenden, und verweisen Sie sie auf den neuen XenDesktop 7.1-Delivery Controller.
11. Aktualisieren Sie alle VDAs auf die entsprechende VDA-Software:
 - Windows 7-VDAs:
 - Führen Sie ein Upgrade auf XenDesktop 7-VDAs durch, wie unter [Durchführen eines Upgrades von Virtual Desktop Agent auf einer VM oder einem Blade-Computer](#) beschrieben.
 - Nach dem Upgrade der VDA-Software auf XenDesktop 7 müssen Sie für Maschinen in einem Katalog oder einer Bereitstellungsgruppe den Katalog und die Bereitstellungsgruppen aktualisieren, wie unter [Aktualisieren eines Maschinenkatalogs](#) und [Aktualisieren einer Bereitstellungsgruppe](#) beschrieben.
 - Aktualisieren Sie VDAs, die unter Windows XP oder Windows Vista ausgeführt werden auf XenDesktop 5.6 FP1, wie unter [Installieren einer früheren Version von Virtual Desktop Agent unter Windows XP oder Windows Vista](#) beschrieben.
12. Beenden Sie den Wartungsmodus für Bereitstellungsgruppen.
13. Konfigurieren Sie StoreFront, um die zuvor vom Webinterface bereitgestellten Desktops zur Verfügung zu stellen. Weitere Informationen zum Installieren und Konfigurieren von StoreFront finden Sie unter [Installieren und Einrichten von StoreFront](#).

Nicht alle XenDesktop 4-Komponenten werden in diesem Release unterstützt. In dieser Tabelle werden die Komponenten aufgeführt, die nicht migriert werden.

Datentyp	Hinweise
Virtual Delivery Agent	Bevor Sie mit einem XenDesktop 7.x-Delivery Controller virtuelle Desktops von

Datentyp	Hinweise XenDesktop 4 verwalten können, müssen Sie ein Upgrade der VDAs auf mindestens XenDesktop 5.x durchführen. Weitere Informationen zum Upgrade der VDAs finden Sie unter Aufgaben nach der Migration.
Controller	Sie müssen neue Controller-Server bereitstellen. Sie können einen XenDesktop 4-Controller nicht auf eine XenDesktop 7.x-Site aktualisieren. XenDesktop 7.x-Sites können keiner XenDesktop 4-Farm und XenDesktop 4-Controller können keiner XenDesktop 7.x-Site beitreten. Darüber hinaus stellt jede Version andere Anforderungen an den Server. XenDesktop 4 erfordert Microsoft Windows Server 2003 und XenDesktop 7.x erfordert Microsoft Windows Server 2008 R2 oder Microsoft Windows Server 2012. XenDesktop 7.1 und 7.5 unterstützen auch Microsoft Windows Server 2012 R2.
Webinterface	Citrix empfiehlt die Verwendung von StoreFront mit XenDesktop 7.x. Weitere Informationen zum Installieren und Konfigurieren von StoreFront finden Sie unter Installieren und Einrichten von StoreFront. Wenn das Installationsprogramm das Webinterface erkennt, wird StoreFront zwar installiert, das Webinterface jedoch nicht entfernt.
Konfiguration der Active Directory-Organisationseinheit	Die Freigabe einer Organisationseinheit zwischen zwei Farmen oder zwei Sites oder einer Farm und einer Site wird nicht unterstützt. Wenn Sie die neue Site mit der Active Directory-basierten Controller-Discovery statt der standardmäßigen registrierungsbasierten Controller-Discovery konfigurieren, müssen Sie eine neue Organisationseinheit hierfür erstellen.
XML-Datei PortICAConfig	Wenn Sie die Standardeinstellungen für diese Datei geändert haben, müssen Sie diese Einstellungen für die neue Site unter Umständen über Gruppenrichtlinienobjekte konfigurieren.
Über XenDesktop 4 Service Pack 1 bereitgestellte Einstellungen für die Konfigurationsprotokollierung	
Bereitstellen dienstbezogener Daten	
Anwendungen	

Datentyp	Hinweise
Liste der Controller	
NetScaler Gateway	
Drosselungseinstellungen für das Ereignisprotokoll	

Details zum Datenimport und Datenexport

Sep 29, 2015

Führen Sie zur Migration von XenDesktop 4 die folgenden allgemeinen Schritte durch:

- Exportieren von Daten aus XenDesktop 4 und Importieren in dieses Release
- Exportieren und Importieren von Benutzerrichtlinieneinstellungen

Nicht alle Daten und Einstellungen werden exportiert. Die folgenden Konfigurationselemente werden nicht migriert. Sie werden zwar exportiert, aber nicht importiert:

- Administratoren
- Einstellungen für delegierte Administration
- Desktopgruppenordner
- Lizenzierungskonfiguration
- Registrierungsschlüssel

Weitere Informationen über Komponenten, die nicht migriert werden, finden Sie unter [Nicht migrierte XenDesktop 4-Komponenten](#).

Folgende Anwendungsfälle werden bei der Migration nicht direkt unterstützt:

- Zusammenführen von Richtlinieneinstellungen oder Desktopgruppen- oder Hostingeinstellungen
- Zusammenführen von privaten Desktops in zufällige Bereitstellungsgruppen
- Anpassen vorhandener Komponenteneinstellungen mit Migrationstools

In dieser, wenn auch nicht vollständigen, Tabelle wird beschrieben, was während der Migration zu diesem Release mit den wichtigsten Daten geschieht.

Datentyp	Exportiert?	Importiert?	Hinweise
Desktopgruppen	J	J	Desktopgruppen werden in diesem Release zu Bereitstellungsgruppen. Symbole von Desktopgruppen werden nicht exportiert. SecureIcaRequired ist auf True gesetzt, wenn DefaultEncryptionLevel in XenDesktop 4 nicht auf "Basic" festgelegt ist. Wenn eine Desktopgruppe in der XenDesktop 4-Farm den gleichen Namen wie eine Bereitstellungsgruppe in der XenDesktop 7.x-Site hat, können Sie Desktops der XenDesktop 4-Gruppe einer Bereitstellungsgruppe mit demselben Namen in der Zielsite hinzufügen. Hierzu müssen Sie den Parameter MergeDesktops angeben, wenn Sie das Importtool ausführen. Die Einstellungen der XenDesktop 7.x-Bereitstellungsgruppe werden nicht mit den Einstellungen der XenDesktop 4-Gruppe überschrieben. Wenn dieser Parameter nicht angegeben wird und eine Gruppe mit demselben Namen wie eine in der XML-Datei definierte Gruppe existiert, zeigt das Tool einen Fehler an und wird angehalten, bevor Daten importiert werden.
Desktops	J	J	Sie können keine privaten Desktops zu zufälligen Bereitstellungsgruppen hinzufügen. Zufällige Desktops können einer statischen Bereitstellungsgruppe nicht hinzugefügt werden.

Datentyp	Exportiert?	Importiert?	Hinweise
Maschinen	J	J	Maschinen werden in vier Maschinenkataloge importiert. Die folgenden Maschinenkataloge werden vom Importtool in der XenDesktop 7.x-Site automatisch erstellt: • Importierte bestehende zufällige (für gepoolte VMs) • Importierte bestehende statische (für zugewiesene VMs) • Importierte physische zufällige (für gepoolte oder Blade-PC) • Importierte physische statische (für private oder Blade-PC) Alle folgenden Importe von Maschinen verwenden dieselben vier Maschinenkataloge.
Poolverwaltungspools	J	J	Umfasst aus mehreren Pools bestehende Pools und Einstellungen für betriebsbereite Pools, einschließlich Zeitplan. PeakBufferSizePercent ist standardmäßig auf 10 % gesetzt. OffPeakBufferSizePercent ist standardmäßig auf 10 % gesetzt. Alle nicht ausgewählten Tage in der Einstellung Business days in XenDesktop 4 werden in dieses Release als Teil des Wochenend-Energieschemas importiert. HostingXD4-Aktionszeiten werden auf die nächste volle Minute gerundet. Startzeiten werden auf die nächste volle Stunde gerundet. Endzeiten werden auf die nächste volle Stunde gerundet.
Farmeinstellungen	J	J	Die folgenden Farmeinstellungen werden als Maschinenrichtlinie importiert: • IcaKeepAlive • AutoClientReconnect • SessionReliability Die Einstellung zum Aktivieren von Flash Player wird nicht importiert.
Richtlinien	J	J	Bestimmte Richtliniendaten werden importiert. Filter, Einstellungen und Drucker werden als Benutzerrichtlinien importiert. Weitere Informationen zum Export und Import von Benutzerrichtlinien finden Sie in der anderen Tabelle in diesem Abschnitt. • Neue Regeln für Zugriffsrichtlinien werden basierend auf XenDesktop 4-Gruppeneinstellungen erstellt. • Beim Import von Richtlinien wird deren relative Prioritätsreihenfolge beibehalten. Sie werden jedoch immer mit einer höheren Priorität als alle auf der XenDesktop 7.x-Site vorhandenen Richtlinien hinzugefügt. • Das Zusammenführen von Richtlinien wird nicht unterstützt. Es gibt keine Möglichkeit, Richtlinien in Active Directory zu importieren. Sie werden immer in der Site gespeichert.
Benutzerzuweisungen	J	J	

Parameter	Exportiert?	Importiert?	Hinweise
Hypervisor-Einstellungen			Dieser Parameter ist erforderlich. Hypervisoradressen werden exportiert, nicht aber die Anmeldeinformationen, die für den Zugriff auf diese Hypervisoren erforderlich sind. Zum Erstellen von Hypervisorverbindungen in der XenDesktop 7.x-Site müssen Sie die Adressen aus der XML-Datei extrahieren und eine PowerShell-Hashtabelle anlegen, in der die Adressen den entsprechenden Anmeldeinformationen zugeordnet sind. Anschließend geben Sie diese Tabelle im Parameter HypervisorConnectionCredentials des Importtools an. Weitere Informationen zum Erstellen der Tabelle finden Sie unter Importieren von XenDesktop 4-Daten. Das Zusammenführen oder Aktualisieren von Hypervisoreinstellungen für vorhandene Desktopgruppen und Hypervisorverbindungen wird nicht unterstützt.
Administratoren	J	N	Es werden keine Administratordaten importiert, einschließlich Daten über delegierte Administratoren. Sie müssen neue Administratoren für die XenDesktop 7.x-Site erstellen.
Lizenzierungskonfiguration	J	N	Enthält Informationen wie Lizenzservername und Edition. Lizenzdateien werden nicht exportiert.
Desktopgruppenordner	J	N	Dieses Release unterstützt keine Desktopgruppenordner. Bei doppelten Desktopgruppennamen (weil unterschiedliche Ordner in der XenDesktop 4-Farm Gruppen mit identischen Namen enthielten) wird das Importtool angehalten, wenn Sie die Namen in der XML-Datei nicht bearbeiten.
Registrierungsschlüssel	J	N	Weitere Informationen zur Implementierung von Registrierungsschlüsseln finden Sie unter Aufgaben nach der Migration .

In dieser Tabelle wird beschrieben, wie Benutzerrichtliniendaten importiert und exportiert werden. Einzelheiten zu Richtlinien finden Sie unter [Referenz für Richtlinienereinstellungen](#).

Kategorie und Einstellung in XenDesktop 4	XML-Datei	Kategorie und Einstellung in XenDesktop 7.x
Bandbreite\Visuelle Effekte\Sitzungslimits OEM-Virtuelle Kanäle	ClientOEMVCBandwidth	Nicht importiert
Clientgeräte\Ressourcen\Sonstiges OEM-Virtuelle Kanäle deaktivieren	DisableOEMVirtualChannels	Nicht importiert
Benutzerarbeitsbereich\Zeitzone Lokale Zeit des Clients nicht verwenden	DoNotUseClientLocalTime	Nicht importiert

Sicherheit\Verschlüsselung Kategorie und Einstellung in XenDesktop 4 SecureICA-Verschlüsselung	ClientSecurityRequirement XML-Datei	Nicht importiert Kategorie und Einstellung in XenDesktop 7.x
Bandbreite\SpeedScreen Beschleunigung der Bildanzeige durch verlustreiche Komprimierung	LossyCompression settings	ICA\Visuelle Anzeige\Festbilder Grad der verlustreichen Komprimierung Schwellenwert für verlustreiche Komprimierung Heavyweight-Komprimierung ICA\Visuelle Anzeige\Bewegtbilder Grad der progressiven Komprimierung Schwellenwert für progressive Komprimierung
Bandbreite\Visuelle Effekte Desktothintergrund deaktivieren	TurnOffWallpaper	ICA\Desktopbenutzeroberfläche Desktothintergrund
Bandbreite\Visuelle Effekte Menüanimation	TurnOffMenuWindowAnimation	ICA\Desktopbenutzeroberfläche Menüanimation
Bandbreite\Visuelle Effekte Fensterinhalt beim Verschieben nicht anzeigen	DoNotShowWindowContentsWhileDragging	ICA\Desktopbenutzeroberfläche Fensterinhalt beim Verschieben anzeigen
Bandbreite\Visuelle Effekte\Sitzungslimits Audio	ClientAudioBandwidth__AllowedBandWidth	ICA\Bandbreite Bandbreitenlimit für die Audioumleitung
Bandbreite\Visuelle Effekte\Sitzungslimits Zwischenablage	ClientClipboardBandwidth__AllowedBandWidth	ICA\Bandbreite Bandbreitenlimit für Zwischenablagenumleitung
Bandbreite\Visuelle Effekte\Sitzungslimits COM-Ports	ClientComBandwidth__AllowedBandWidth	Die COM-Portumleitung ist in XenDesktop 7.x veraltet.
Bandbreite\Visuelle Effekte\Sitzungslimits Laufwerke	ClientDriveBandwidth__AllowedBandWidth	ICA\Bandbreite Bandbreitenlimit für Dateiumleitung

Bandbreite\Visuelle Effekte\Sitzungslimits Kategorie und Einstellung in XenDesktop 4 LPT-Ports	ClientLptBandwidth__AllowedBandWidth XML-Datei	Die LPT-Portumleitung ist in XenDesktop 7.x veraltet. Kategorie und Einstellung in XenDesktop 7.x
Bandbreite\Visuelle Effekte\Sitzungslimits Sitzung insgesamt	OverallBandwidth__AllowedBandWidth	ICA\Bandbreite Bandbreitenlimit für Sitzung insgesamt
Bandbreite\Visuelle Effekte\Sitzungslimits Drucker	LimitPrinterBandWidth__AllowedBandWidth	ICA\Bandbreite Bandbreitenlimit für Druckerumleitung
Clientgeräte\Ressourcen\Audio Mikrofone	ClientAudioMicrophone__TurnOn	ICA\Audio Clientmikrofonumleitung
Clientgeräte\Ressourcen\Audio Tonqualität	ClientAudioQuality__Quality	ICA\Audio Audioqualität
Clientgeräte\Ressourcen\Audio Lautsprecher deaktivieren	DisableClientAudioMapping	ICA\Audio Clientaudioumleitung
Clientgeräte\Ressourcen\Laufwerke Verbindung (Connection)	ConnectClientDriveAtLogon__TurnOn	ICA\Dateiumleitung Laufwerke automatisch verbinden
Clientgeräte\Ressourcen\Laufwerke Diskettenlaufwerke deaktivieren	DisableClientDriveMapping__DisableFloppyDrive	ICA\Dateiumleitung Clientdiskettenlaufwerke
Clientgeräte\Ressourcen\Laufwerke Festplatten deaktivieren	DisableClientDriveMapping__DisableHardDrive	ICA\Dateiumleitung Lokale Clientfestplattenlaufwerke
Clientgeräte\Ressourcen\Laufwerke CD-ROM-Laufwerke deaktivieren	DisableClientDriveMapping__DisableCdrom	ICA\Dateiumleitung Optische Clientlaufwerke
Clientgeräte\Ressourcen\Laufwerke Remotelaufwerke deaktivieren	DisableClientDriveMapping__DisableRemote	ICA\Dateiumleitung Clientnetzlaufwerke
Clientgeräte\Ressourcen\Laufwerke USB-Laufwerke deaktivieren	DisableClientDriveMapping__DisableUSB	ICA\Dateiumleitung Clientwechsellaufwerke

Clientgeräte\Ressourcen\Laufwerke\Optimierung Kategorie und Einstellung in XenDesktop 4	CDMAsyncWrites XML-Daten	ICA\Datei- und Einstellung in XenDesktop 7.x Asynchrones Schreiben
Asynchrones Schreiben		verwenden
Clientgeräte\Ressourcen\Sonstiges Zwischenablagezuordnung deaktivieren	DisableClientClipboardMapping	ICA Clientzwischenablagenumleitung
Clientgeräte\Ressourcen\Ports COM-Ports deaktivieren	DisableClientCOMPortMapping	Die COM-Portumleitung ist in XenDesktop 7.x veraltet.
Clientgeräte\Ressourcen\Ports LPT-Ports deaktivieren	DisableClientLPTPortMapping	Die LPT-Portumleitung ist in XenDesktop 7.x veraltet.
Clientgeräte\Ressourcen\USB USB	RemoteUSBDevices__DisableRemoteUSBDevices	ICA\USB-Geräte Client-USB-Geräteumleitung
Drucken\Clientdrucker Automatische Erstellung	ConnectClientPrinterAtLogon__Flag	ICA\Drucken\Clientdrucker Clientdrucker automatisch erstellen
Drucken\Clientdrucker Legacy-Clientdrucker	LegacyClientPrinters__TurnOn	ICA\Drucken\Clientdrucker Clientdruckernamen
Drucken\Clientdrucker Speicherung von Druckereigenschaften	ModifiedPrinterProperties__WriteMethod	ICA\Drucken\Clientdrucker Speicherung von Druckereigenschaften
Drucken\Clientdrucker Weiterleiten von Druckaufträgen	ClientPrintingForNetworkPrinter__TurnOn	ICA\Drucken\Clientdrucker Direkte Verbindungen zu Druckservern
Drucken\Clientdrucker Clientdruckerzuordnung deaktivieren	DisableClientPrinterMapping	ICA\Drucken Clientdruckerumleitung
Drucken\Treiber Native Druckertreiber automatisch installieren	PrintDriverAutoInstall__TurnOn	ICA\Drucken\Treiber Automatische Installation von mitgelieferten Druckertreibern
Drucken\Treiber Universeller Treiber	ClientPrintDriverToUse	ICA\Drucken\Treiber Verwendung universeller Druckertreiber

Kategorie und Einstellung in XenDesktop 4	XML-Datei	Kategorie und Einstellung in XenDesktop 7.x
Drucken\Sitzungsdrucker	NetworkPrinters	ICA\Drucken
Sitzungsdrucker		Sitzungsdrucker
Drucken\Sitzungsdrucker	DefaultToMainClientPrinter__NetworkDefault	ICA\Drucken
Standarddrucker des Clients wählen	DefaultToMainClientPrinter__TurnOn	Standarddrucker

Exportieren aus einer XenDesktop 4-Farm

Sep 29, 2015

Das Exporttool "XdExport" extrahiert Daten aus einer einzelnen XenDesktop 4-Farm und generiert eine XML-Datei aus Darstellungen der Datenwerte.

Die exportierten Datentypen werden unter [Details zum Datenimport und Datenexport](#) beschrieben.

Das Schema der XML-Datei wird in einer Datei mit der Bezeichnung XdFarm.xsd bereitgestellt, die im Download des Importtools enthalten ist: XdExport.zip und XdImport.zip.

Führen Sie das Tool auf einer Maschine aus, die in der Farm, aus der Daten exportiert werden, ein XenDesktop 4-Controller ist. Auf dieser Maschine muss das XenDesktop 4-PowerShell SDK installiert sein.

Sie müssen über die folgenden Berechtigungen für den Export der Daten verfügen:

- Die Benutzeridentität eines mindestens lesezugriffsberechtigten Citrix Administrators der Farm
- Berechtigung zum Lesen der Registrierung

Obwohl dies nicht empfohlen wird, können Sie das Tool ausführen, während der XenDesktop Controller aktiv verwendet wird (z. B. wenn Benutzer bei VDAs angemeldet sind).

Citrix empfiehlt dringend Folgendes:

- Update des XenDesktop 4-Controllers, auf dem das Tool ausgeführt wird, mit öffentlichen Hotfixes
- Keine Konfigurationsänderungen an der Site während der Export ausgeführt wird (z. B. Entfernen von Desktopgruppen)

1. Laden Sie XdExport.zip herunter und extrahieren Sie die Dateien auf den XenDesktop 4-Controller.
2. Führen Sie an einer Eingabeaufforderung XdExport.exe aus. Sie können folgende Parameter einbeziehen:

Parameter	Beschreibung
-Verbose	Generiert Nachrichten, die detaillierte Fortschrittsinformationen liefern.
-FilePath	Der Speicherort der XML-Datei, in die die Farmdaten exportiert werden. Standard = .\XdSettings.xml
-Overwrite	Überschreibt jede in -FilePath angegebene Datei. Wenn Sie diesen Parameter nicht angeben und eine Ausgabedatei bereits vorhanden ist, schlägt das Tool mit folgender oder einer ähnlich lautenden Fehlermeldung fehl: Fehler: Datei bereits vorhanden. Geben Sie -Overwrite an, um das Überschreiben der Datei zuzulassen.
-? oder -help	Zeigt eine Beschreibung der Parameter an. Beim Beenden werden keine Daten exportiert.

3. Bei erfolgreicher Ausführung des Tools wird die Meldung Fertig angezeigt. Die Datei XdSettings.xml befindet sich an dem vom Parameter FilePath angegebenen Speicherort.
Schlägt das Tool fehl, wird eine Fehlermeldung angezeigt.

Wenn Sie das Exporttool erfolgreich ausgeführt haben, überprüfen und bearbeiten Sie die XML-Datei wie unter [Bearbeiten der XML-Datei des Migrationstools](#) beschrieben.

Bearbeiten der XML-Datei des Migrationstools

Sep 29, 2015

Vor dem Importieren von Daten in eine XenDesktop 7-Website bearbeiten und prüfen Sie den Inhalt der vom Exporttool erstellten XML-Datei XdExport, insbesondere dann, wenn Sie die Migration in mehreren Phasen durchführen und beim Import einige Benutzer, Bereitstellungsgruppen und Richtlinien anderen vorgezogen haben.

Sie können den Inhalt der Datei mit einem beliebigen Texteditor anzeigen oder ändern. Es besteht aber auch die Möglichkeit, einen speziellen XML-Editor wie Microsoft XML Notepad einzusetzen.

Bestimmte Elemente des XML-Inhalts müssen in der XML-Datei vorhanden sein, damit sie vom Importtool XdImport akzeptiert wird.

Das benötigte XML-Schema ist in der Datei XdFarm.xsd definiert, die im Download des Migrationstools enthalten ist. Berücksichtigen Sie bei der Arbeit mit dieser Datei Folgendes:

- In der Datei gibt ein Attribut minOccurs mit einem Wert von 1 oder höher an, dass bestimmte Elemente vorhanden sein müssen, wenn das übergeordnete Element vorhanden ist.
- Wenn die an das Importtool übergebene XML-Datei ungültig ist, stoppt das Tool und eine Fehlermeldung wird angezeigt, mit deren Hilfe Sie in der XML-Datei nach dem Problem suchen können.

Zum Importieren nur einer Untergruppe von Desktopgruppen und Desktops müssen Sie den Inhalt des Elements DesktopGroups bearbeiten. Das Element DesktopGroups kann viele DesktopGroup-Elemente enthalten und in jedem DesktopGroup-Element befindet sich ein Desktops-Element, das zahlreiche Desktop-Elemente enthalten kann.

Sie dürfen das DesktopGroups-Element nicht löschen. Sie können aber alle DesktopGroup-Elemente löschen und es leer lassen. Ebenso muss in jedem DesktopGroup-Element das Desktops-Element vorhanden sein. Es muss aber keine Desktop-Elemente enthalten.

Löschen Sie Desktop- oder DesktopGroup-Elemente, um zu verhindern, dass bestimmte einzelne Maschinen oder ganze Bereitstellungsgruppen importiert werden. Die XML-Datei enthält beispielsweise:

```
... ..
```

Bei diesem Beispiel wird durch Ihre Änderungen verhindert, dass die Gruppe "Group1" importiert wird. Nur Machine3 aus der Gruppe "Group2" wird importiert:

```
... ..
```

In XenDesktop 4 können Desktopgruppen in Ordnern organisiert werden, Desktopgruppen mit dem gleichen Namen können in verschiedenen Ordnern erscheinen und der interne Name einer Desktopgruppe ist der Name, der den Benutzern angezeigt wird.

In diesem Release hingegen dürfen Bereitstellungsgruppen nicht in Ordnern platziert werden und jede Bereitstellungsgruppe muss einen eindeutigen internen Namen haben, wobei der Name, der den Benutzern angezeigt wird, sich vom internen Namen unterscheiden kann. Aufgrund dieser Unterschiede müssen Sie möglicherweise Desktopgruppen umbenennen.

In Ihrer XenDesktop 4-Farm können Sie beispielsweise zwei verschiedene Desktopgruppen haben, die zwei verschiedenen Benutzern mit dem Namen "Mein Desktop" angezeigt wird. Dies kann durch Desktopgruppenordner ermöglicht werden. Wenn diese Bereitstellungsgruppen in der XenDesktop 7.x-Site getrennt bleiben sollen, müssen Sie die Namen der

Desktopgruppen in der XML-Datei in eindeutige Namen ändern.

Wenn eine Bereitstellungsgruppe in der XenDesktop 7.x-Site über denselben Namen verfügt wie die zu importierende Desktopgruppe und die Bereitstellungsgruppen in der XenDesktop 7.x-Site getrennt bleiben sollen, müssen Sie den Namen der XenDesktop 4-Desktopgruppe in der XML-Datei ändern, damit der Name in der Site eindeutig bleibt. Wenn die zu importierende Desktopgruppe tatsächlich der XenDesktop 7.x-Bereitstellungsgruppe entspricht und die Maschinen in der XML-Datei mit der vorhandenen Desktopgruppe zusammengeführt werden sollen, müssen Sie die Desktopgruppe nicht umbenennen. Geben Sie stattdessen den Parameter -MergeDesktops im Importtool an. Die XML-Datei enthält beispielsweise:

... \Sales ... \Finance

Entfernen Sie die doppelten Namen wie folgt:

... \Sales ... \Finance

Sie können einzelne Richtlinien aus der XML-Datei löschen und eindeutige Namen festlegen, um doppelte Richtlinienamen zu vermeiden. Das Zusammenführen von Richtlinien wird nicht unterstützt.

Informationen zum Import von Richtlinien:

- Beim Import von Richtlinien werden entweder alle Richtlinien erfolgreich importiert oder es werden keine Richtlinien importiert, wenn an irgendeiner Stelle ein Fehler auftritt.
- Der Import zahlreicher Richtlinien mit umfangreichen Einstellungen kann mehrere Stunden dauern.
- Wenn Sie Richtlinien in Batches importieren, können deren ursprüngliche Prioritäten beeinträchtigt werden. Wenn Sie Richtlinien importieren, werden die relativen Prioritäten der importierten Richtlinien beibehalten. Sie erhalten jedoch eine höhere Priorität als bereits in der Site vorhandene Richtlinien. Wenn beispielsweise vier Richtlinien mit den Prioritäten 1 bis 4 in zwei Batches importiert werden sollen, sollten Sie die Richtlinien mit den Prioritäten 3 und 4 zuerst importieren, da der zweiten Gruppe automatisch eine höhere Priorität eingeräumt wird.

Zum ausschließlichen Importieren einer Untergruppe der Richtlinien in die XenDesktop 7.x-Site bearbeiten Sie den Inhalt des Policies-Elements. Das Policies-Element kann zahlreiche Policy-Elemente enthalten. Sie dürfen das Policies-Element nicht löschen. Sie können aber alle Policy-Elemente löschen und es leer lassen. Löschen Sie alle Policy-Elemente, um den Import bestimmter XenDesktop 4-Farmrichtlinien zu vermeiden. Die XML-Datei enthält beispielsweise:

... ...

Wenn Sie den Import von XenDesktop 4-Richtlinien verhindern möchten, um mögliche Konflikte mit bereits in der XenDesktop 7.x-Site konfigurierten Richtlinien zu vermeiden, bearbeiten Sie die Datei, indem Sie einzelne Policy-Elemente wie folgt entfernen:

Alternativ können Sie die Datei so bearbeiten, dass die Richtlinie wie folgt mit einem anderen Namen importiert wird:

... ...

Importieren von XenDesktop 4-Daten

Sep 29, 2015
Das Importtool "XdImport" liest Einstellungen aus XenDesktop 4, die in der vom Exporttool "XdExport" erzeugten XML-Datei enthalten sind, und wendet diese Einstellungen auf eine vorhandene XenDesktop 7.x-Site an. Das Importtool ist das PowerShell-Skript Import-XdSettings.ps1.

Nicht alle Daten werden aus der Datei XdSettings.xml importiert. Informationen zu den Datentypen, die importiert werden, finden Sie unter [Details zum Datenimport und Datenexport](#).

Um nur eine Teilmenge der exportierten Daten anzuwenden, bearbeiten Sie die XML-Datei, bevor Sie das Importtool ausführen. Sie können beispielsweise Desktopgruppen und Richtlinien entfernen, die in der XenDesktop 7.x-Bereitstellung nicht benötigt werden.

Das Importtool wird erfolgreich ausgeführt, wenn Sie ganze Elemente leer lassen. Sie können beispielsweise alle Desktopgruppen problemlos löschen. Das Tool überprüft vor dem Import der Daten immer die XML-Datei. Weitere Informationen zum Bearbeiten der XML-Datei finden Sie unter [Bearbeiten der XML-Datei des Migrationstools](#).

Unter [Details zum Datenimport und Datenexport](#) wird beschrieben, wie die Daten der XML-Datei in die XenDesktop 7.x-Site migriert werden.

Führen Sie XdImport auf jeder Maschine aus, auf der alle XenDesktop 7.x-SDKs installiert sind. Sie müssen als Volladministrator für XenDesktop angemeldet sein, um das Tool auszuführen.

Stellen Sie vor dem Importieren sicher, dass Sie eine XenDesktop 7.x-Site einschließlich Datenbank eingerichtet haben. Vollständige Angaben zu den Anforderungen der Datenbank finden Sie im Abschnitt "Datenbank" unter [Systemanforderungen für XenDesktop 7.1](#).

Citrix empfiehlt, den Import in XenDesktop 7.x durchzuführen, bevor Benutzertests oder eine allgemeine Sitekonfiguration durchgeführt werden. Führen Sie Konfigurationen nur zusammen, wenn die Site nicht verwendet wird.

1. Erstellen Sie eine XenDesktop 7.x-Site.
2. Laden Sie die Datei XdImport.zip herunter und extrahieren Sie die Dateien auf die Maschine, auf der das Tool ausgeführt wird.
3. Führen Sie in einer PowerShell-Sitzung Import-XdSettings.ps1 aus. Sie können folgende Parameter angeben:

Parameter	Beschreibung
- HypervisorConnectionCredentials	Dieser Parameter ist erforderlich. Eine PowerShell-Hashtabelle, in der PS-Credential-Instanzen Hypervisoradressen als Voraussetzung für die Erstellung von Hypervisorverbindungen zugeordnet werden. Standard = @{} Geben Sie die Anmeldeinformationen für den Hypervisor ein, mit dem die XenDesktop 4-Farm eine Verbindung herstellt. Für einen einzelnen Hypervisor können Sie das Argument folgendermaßen erstellen: \$credential = Get-Credential \$mappings = @{ "http://" = \$credential } .\Import-XdSettings.ps1 -FilePath .\XdSettings.xml -HypervisorConnectionCredentials \$mappings Beachten Sie, dass die in der Hashtabelle angegebene Adresse mit der Adresse in der XML-Datei genau übereinstimmen muss. Beispiel: Erstellen Sie mit einem XenServer- und einem VMware-Hypervisor folgendes Argument: \$Xencredential = Get-Credential \$VMWcredential = Get-Credential \$mappings = @{ "http://" = \$Xencredential; "http://SDK" = \$VMWcredential } .\Import-XdSettings.ps1 -FilePath .\XdSettings.xml -HypervisorConnectionCredentials \$mappings
-FilePath	Der Speicherort der XML-Datei, aus der die Farmdaten importiert werden. Der Wert für ist erforderlich.
-AdminAddress	Der Name des Controllers in der XenDesktop 7.x-Site. Standard = localhost
-MergeDesktops	Fügt in der XML-Datei definierte Desktops Bereitstellungsgruppen in der XenDesktop 7.x-Site hinzu, deren Namen mit Gruppen in der XML-Datei übereinstimmen hinzugefügt. Wird dieser Parameter nicht angegeben, wird vorhandenen Desktopgruppen in der XenDesktop 7.x-Site kein Inhalt hinzugefügt.
-SkipMachinePolicy	Das Skript erstellt keine Maschinenrichtlinie mit Einstellungen auf Siteebene. Wenn Sie diesen Parameter nicht angeben und die Maschinenrichtlinie für die Site vorhanden ist, schlägt das Skript fehl.
-WhatIf	Führt einen Testlauf durch, um festzustellen, was an der XenDesktop 7.x-Site geändert und was ihr hinzugefügt werden würde. Wenn dieser Parameter angegeben wird jedoch nicht geändert.
-LogFilePath	Der vollständige Pfad der Protokolldatei. Das Protokoll enthält Text, der alle auf der XenDesktop 7.x-Site durchgeführten Schreibvorgänge beschreibt. Standard = .\Import-XdSettings.log
-? oder -help	Zeigt Text an, der die Parameter beschreibt. Beim Beenden werden keine Daten importiert.

Wenn die XML-Datei Richtliniendaten enthält, werden alle Richtlinien erfolgreich importiert. Tritt jedoch an irgendeiner Stelle ein Fehler auf, werden keine Richtliniendaten importiert. Der Import zahlreicher Richtlinien mit umfangreichen Einstellungen kann mehrere Stunden dauern.

4. Nach Abschluss des Skripts wird die Meldung Fertig angezeigt.

Nach erfolgreichem Import der Daten aus der XML-Datei können Sie weitere Import- und Exportvorgänge durchführen. Wenn Sie jedoch alle relevanten Daten importiert haben, können Sie die unter [Aufgaben nach der Migration](#) beschriebenen Aufgaben nach der Migration durchführen.

Aufgaben nach der Migration

Sep 29, 2015

Nach dem erfolgreichen Importieren von Daten aus einer XenDesktop 4-Farm in eine XenDesktop 7.x-Site müssen Sie folgende Aufgaben durchführen, bevor Sie die neue Site als Produktionsumgebung nutzen können:

- Durchführen eines Upgrades der Virtual Delivery Agents (VDAs) der Maschinen. Obwohl es nicht erforderlich ist, empfiehlt Citrix, dass Sie zunächst ein Upgrade der VDAs und dann erst ein Upgrade der Controller, von Studio oder von Director durchführen. Weitere Informationen finden Sie unter [Direktes Upgrade](#).
 - Windows Vista und Windows XP: Upgrade auf XenDesktop 5.6 Feature Pack 1 Virtual Desktop Agent
 - Windows 7: Upgrade auf XenDesktop 7.x Virtual Delivery Agent.
- Erstellen Sie alle Administratoren, die für die XenDesktop 7.x-Site benötigt werden.
- Citrix empfiehlt, ein Update der Benutzergeräte auf die aktuelle Version von Citrix Receiver durchzuführen, um von Hotfixes zu profitieren und Unterstützung für die aktuellen Funktionen zu erhalten. Weitere Informationen finden Sie unter [Receiver und Plug-Ins](#).
- Ändern Sie die importierten Desktops so, dass die registrierungsbasierte Controller-Discovery verwendet wird, und verweisen Sie sie auf die XenDesktop 7.x-Controller mit einer der folgenden Methoden:
 - Bearbeiten Sie die Registrierung manuell:
 - Entfernen Sie nicht benötigte Organisationseinheit-GUID-Registrierungseinträge.
 - Fügen Sie einen Registrierungseintrag ListOfDDCs hinzu.
 - Richten Sie eine Maschinenrichtlinie ein, um die Liste der Controller auf die Desktops mit der Active Directory-Richtlinie GPMC.msc zu verteilen. Sie können diese Einstellung nicht über Studio konfigurieren.

Die registrierungsbasierte Controller-Discovery ist die Standardeinstellung für XenDesktop 7.x. Die Active Directory-basierte Discovery steht aber weiterhin zur Verfügung. Weitere Informationen finden Sie unter [Überlegungen zu Active Directory](#).
- Optional können Sie die folgenden Registrierungsschlüsseleinstellungen implementieren, die im Abschnitt zu den bewährten Methoden für die registrierungsbasierte Registrierung in XenDesktop unter [CTX133384](#) beschrieben werden:
 - HeartbeatPeriodMS
 - PrepareSessionConnectionTimeoutSec
 - MaxWorkers
 - DisableActiveSessionReconnect
 - ControllersGroupGuid

Wenn Sie dies unterlassen, werden die XenDesktop 7.x-Standardeinstellungen für diese Schlüssel verwendet.
- Heben Sie den Wartungsmodus für die importierten Maschinen auf, wenn sie vor dem Generieren der XML-Datei in XenDesktop 4 im Wartungsmodus ausgeführt wurden.
- Überprüfen Sie die XenDesktop 7.x-Einstellungen, um sicherzustellen, dass sie korrekt sind, insbesondere dann, wenn Sie die XML-Datei PortICAConfig unter XenDesktop 4 geändert haben.
- Überprüfen Sie alle unter [Details zum Datenimport und Datenexport](#) beschriebenen Migrationskomponenten, um sicherzustellen, dass die Migration erfolgreich abgeschlossen wurde.

Desktop- und Anwendungsbereitstellung

Sep 29, 2015

Citrix Studio bietet mehrere Methoden für die Bereitstellung von Desktops und Anwendungen. Mit diesen Optionen können Sie die Computinganforderungen aller Typen von Benutzern in der Organisation erfüllen.

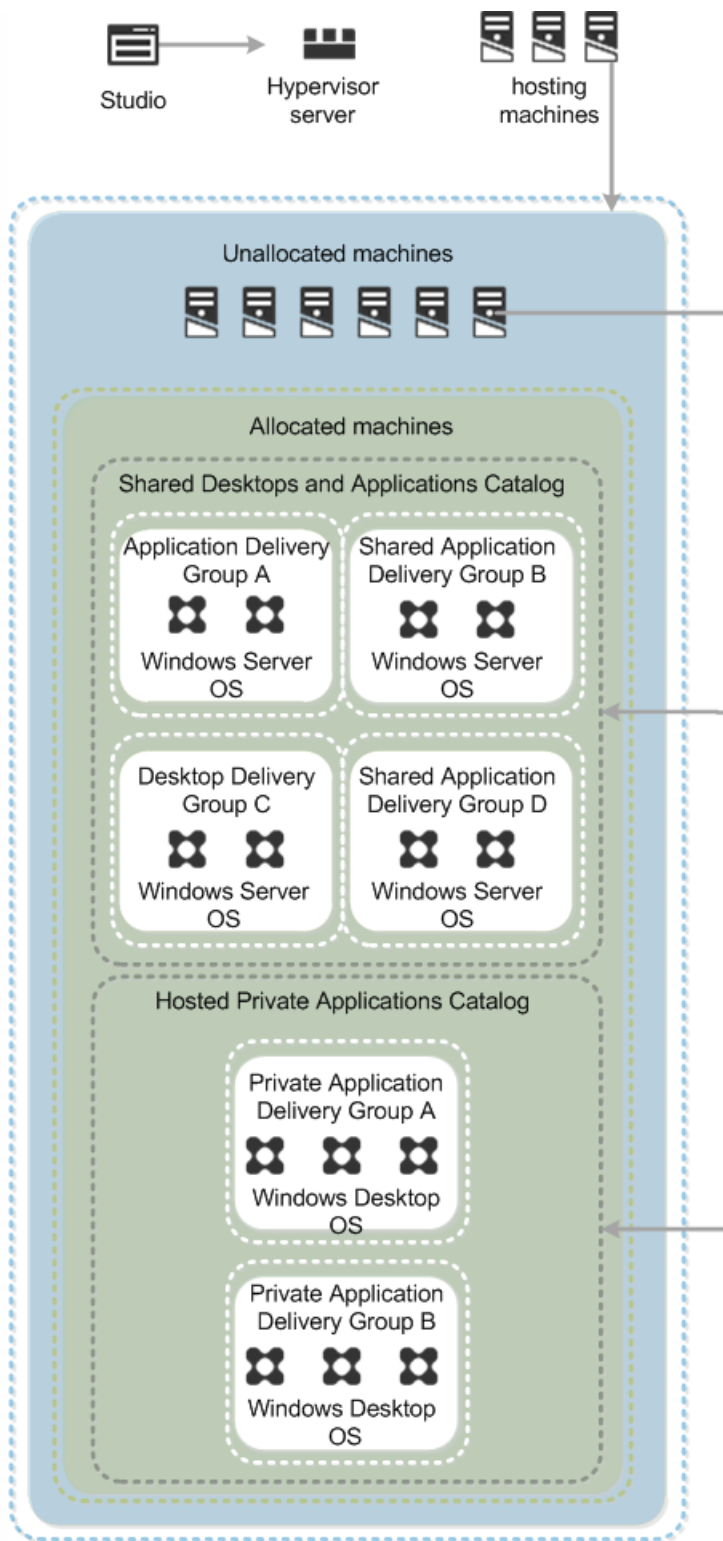
Sammlungen von identischen virtuellen Maschinen (VMs) oder von physischen Computern werden als eine Einheit verwaltet, die als Maschinenkatalog bezeichnet wird. Wenn die Bereitstellung nicht mit Provisioning Services oder Maschinenerstellungsdiensten erfolgt, empfiehlt Citrix, dass Sie Microsoft System Center Configuration Manager oder eine Anwendung eines Drittanbieters verwenden, um sicherzustellen, dass die Maschinen in dem Katalog konsistent sind. Informationen zur Verwendung von Configuration Manager mit XenApp bzw. XenDesktop finden Sie unter [CitrixConnector 7.5 für System Center Configuration Manager 2012](#).

Die Maschinen in diesem Maschinenkatalog sind entweder für ein Windows-Desktop- oder ein Windows-Serverbetriebssystem konfiguriert. Für Benutzer, die auf Bürocomputer remote zugreifen, können Sie Remote-PC-Zugriff für die Maschinenkataloge auswählen.

Hinweis: Amazon Web Services (AWS) unterstützt nur Maschinenkataloge für Serverbetriebssystemmaschinen und [Server-VDI](#).

Sie erstellen Bereitstellungsgruppen, um Desktops, Anwendungen oder eine Kombination aus beidem einem oder mehreren Benutzern bereitzustellen. Mit einer Bereitstellungsgruppe ordnen Sie Maschinen und Anwendungen eines Maschinenkatalogs flexibel Benutzern zu.

Die folgende Abbildung zeigt die Maschinenkataloge und Bereitstellungsgruppen von Studio.



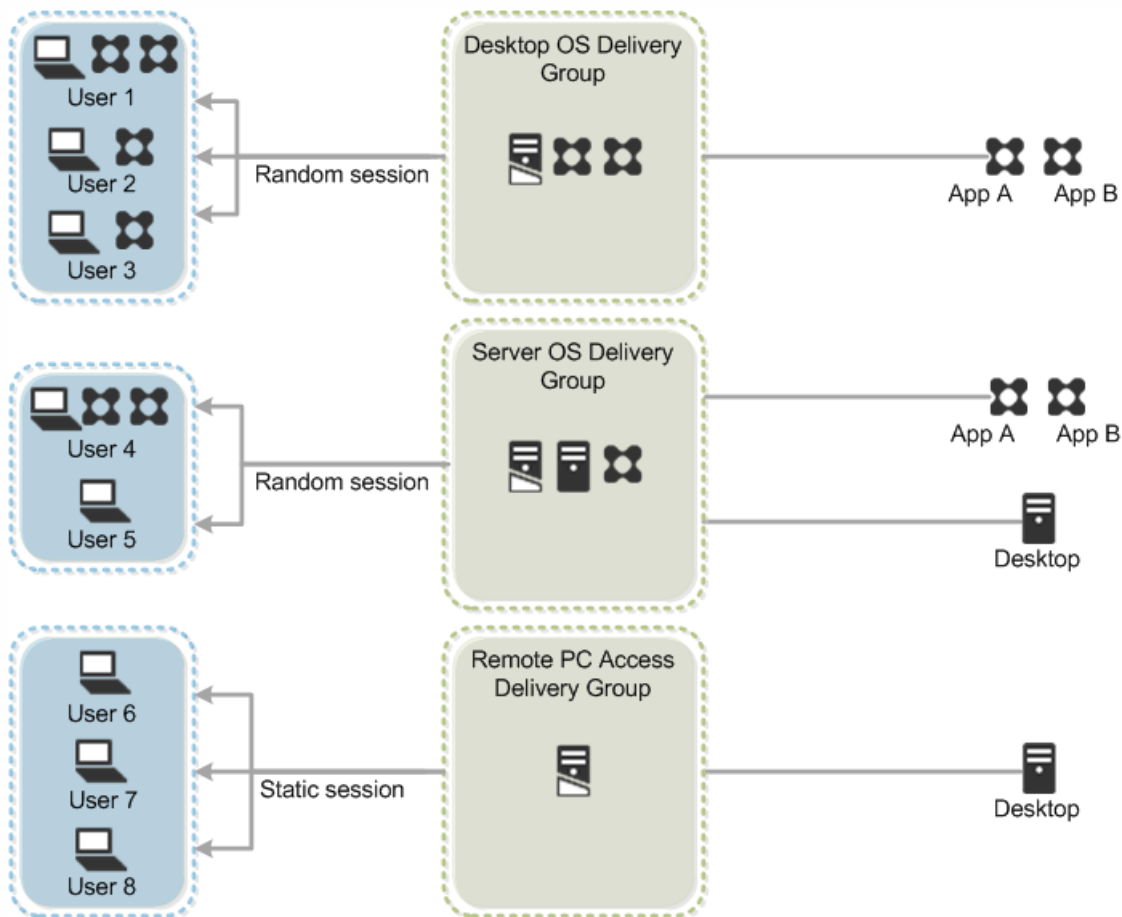
Bereitstellungsgruppen ermöglichen Folgendes:

- Verwenden von Maschinen aus mehreren Maschinenkatalogen
- Zuweisen eines Benutzers zu mehreren Maschinen
- Zuweisen mehrerer Benutzer zu einer Maschine

Im Rahmen der Erstellung geben Sie die folgenden Bereitstellungsgruppeneigenschaften an:

- Den Bereitstellungsgruppen zugewiesene Benutzer, Gruppen und Anwendungen
- Desktopeinstellungen, die die Benutzeranforderungen erfüllen
- Energieverwaltungsoptionen für Desktops

Die folgende Abbildung zeigt, wie Benutzer über Maschinenkataloge und Bereitstellungsgruppen auf Desktops und Anwendungen zugreifen.



Es gibt mehrere Methoden für das Bereitstellen von Desktops und Anwendungen. Welche Methode Sie wählen, hängt von den gehosteten Anwendungstypen ab, den verfügbaren Systemressourcen und welche Art von Benutzererfahrung Sie bereitstellen möchten.

Ausführliche Informationen zum Auswählen einer Bereitstellungsmethode finden Sie unter [Wählen einer Anwendungs- und Desktop-Bereitstellungsmethode](#).

Anhand der folgenden Informationen können Sie eine Entscheidung für eine Bereitstellungsmethode treffen:

- Serverbetriebssysteme bieten Zugriff für mehrere Benutzer mit Verbindung zu einer Maschine oder einer Anwendung.
- Desktopbetriebssysteme bieten Zugriff für einzelne Benutzer mit Verbindung zu einer Maschine oder Anwendung.
- Remote-PC-Zugriff bietet Zugriff auf Bürocomputer von anderen Standorten aus über eine Einzelbenutzerverbindung. Bei Remote-PC-Zugriff wird VPN nicht für die Sicherheit benötigt.

Einrichten von Maschinenkatalogen

Sep 29, 2015

Um Benutzern Desktops und Anwendungen bereitzustellen, erstellt der Maschinenadministrator einen Katalog von Maschinen. Der Zuordnungsadministrator teilt Benutzern Maschinen aus dem Maschinenkatalog zu, indem er Bereitstellungsgruppen erstellt.

Per Cloud bereitgestellte Maschinenkataloge

In diesem Release wird das Hosten einer Site in einer Cloudumgebung unterstützt, die mit Citrix CloudPlatform oder Amazon Web Services (AWS) verwaltet wird. Berücksichtigen Sie beim Erstellen von per Cloud bereitgestellten Maschinenkatalogen Folgendes:

- Maschinen werden von Maschinenerstellungsdiensten (MCS) bereitgestellt.
- AWS unterstützt nur Maschinenkataloge für Serverbetriebssystemmaschinen und [Server-VDI](#).
- AWS unterstützt keine Remote-PC-Zugriff-Kataloge.

Vorbereiten der Maschinenkatalogerstellung

Bevor Sie beginnen, sollten Sie Folgendes sicherstellen:

- Identifizieren Sie anhand Ihrer Umgebung und der Benutzer die zu erstellenden Maschinenkatalogtypen. Weitere Informationen finden Sie unter [Desktop- und Anwendungsbereitstellung](#).
- Identifizieren Sie die Maschineninfrastruktur der Maschinenkataloge für Ihre Desktopbereitstellung:
 - Mit Energieverwaltung
 - Dieser Maschinenkatalog enthält Maschinen, deren Energieverwaltung über Studio erfolgt. Wenn beim Erstellen des Maschinenkatalogs keine Hypervisor-Verbindungen konfiguriert werden, ist diese Option nicht verfügbar.
 - Verbindungen mit einer Cloudumgebung erfolgen immer über per MCS bereitgestellte virtuelle Maschinen oder vorhandene Images, die einen anderen Dienst oder eine andere Technologieoption verwenden.
 - Ohne Energieverwaltung: Dieser Maschinenkatalog enthält Maschinen (normalerweise physische Computer), bei denen keine Energieverwaltung über Studio erfolgt.
- Ziehen Sie die Art der Verwaltung und der Erstellung von Maschinen in Betracht. Sie wirkt sich auf den empfohlenen Maschinenkatalogtyp aus. Sie verwalten Maschinenimages eines Maschinenkatalogs mit einer der folgenden Methoden:
 - Die Maschinenerstellungsdienste verwalten die virtuellen Maschinen mit einem Masterimage bzw. einer Vorlage in der Umgebung, sodass Sie Zielgeräte problemlos über ein Masterimage bzw. eine Vorlage verwalten und aktualisieren können.
 - Provisioning Services verwalten Zielgeräte als Gerätesammlung. Eine von einem Masterzielgerät per Image erstellte Provisioning Services-vDisk stellt die Desktops und Anwendungen bereit und ermöglicht die Nutzung der Rechenleistung der vorhandenen Hardware bzw. virtuellen Maschinen.
 - Bei vorhandenen Images wird ein anderer Dienst oder eine andere Technologieoption verwendet. Sie verwalten und stellen Desktops und Anwendungen bereit, die bereits auf VMs im Datacenter migriert wurden. Sie müssen die Zielgeräte individuell oder kollektiv mit ESD-Tools (Electronic Software Distribution) verwalten.
- Vorbereiten der Umgebung:
 - Bereiten Sie ein Masterimage oder eine Vorlage vor.
 - Geben Sie Active Directory-Computerkonten an.
 - Netzwerkkartenkonfiguration: Wenn mehrere Netzwerkkarten in einem Masterimage verfügbar sind, können Sie auswählen, welche mit dem Maschinenkatalog verwendet wird, und Sie können der Netzwerkkarte auch ein

verfügbares Netzwerk zuweisen. Sie können Netzwerkkarten auch wie unter [Konfigurieren von Netzwerkkarten](#) beschrieben hinzufügen.

Erstellen von Maschinenkatalogen

Erstellen Sie Maschinenkataloge mit dem Assistenten zum Erstellen von Maschinenkatalogen. Sie müssen im Assistenten Angaben zu Ihrer Umgebung machen und erhalten auf der Basis Ihrer Antworten Hilfe bei der Auswahl des für Ihre Anforderungen geeigneten Maschinenkatalogtyps.

Testen des Maschinenkatalogs

Nach dem Erstellen oder Ändern des Maschinenkatalogs können Sie Tests ausführen, um sicherzustellen, dass die Maschinenkataloge nun einsatzbereit sind.

Alle Maschinen in einem Maschinenkatalog benötigen ein entsprechendes Active Directory-Computerkonto. Wenn Sie zufällige oder statische Maschinenkataloge erstellen möchten und Zugriff auf ein Active Directory-Domänenadministratorkonto haben, können neue Konten beim Erstellen des Maschinenkatalogs von Studio erstellt werden. Wenn Sie nicht über die notwendigen Berechtigungen verfügen, stellen Sie sicher, dass für die Maschinen genügend nicht verwendete Active Directory-Computerkonten zur Verfügung stehen, bevor Sie die Aufgabe Maschinenkatalog erstellen starten.

Sie können die zu verwendenden vorhandenen Computerkonten auswählen, indem Sie beim Erstellen des Katalogs Active Directory durchsuchen. Alternativ dazu können Sie eine CSV-Datei mit einer Liste von Kontonamen importieren. Studio erfordert das folgende Format für Computerkonten, die aus CSV-Dateien importiert werden.

[ADComputerAccount] ADcomputeraccountname.domain ...

Bei vorhandenen und physischen Maschinenkatalogen wählen Sie vorhandene Konten aus oder importieren diese und weisen jeder VM oder physischen Maschine ein Active Directory-Computerkonto und ein Benutzerkonto zu. Bei Maschinen, die mit Provisioning Services erstellt werden, werden die Active Directory-Computerkonten für Zielgeräte mit Provisioning Services und vorhandenen Active Directory-Tools verwaltet. Weitere Informationen zur Active Directory-Integration mit Provisioning Services finden Sie unter [Provisioning Services](#).

Weitere Informationen zur Verwaltung von Active Directory-Konten bzw. Organisationseinheitskonten eines Maschinenkatalogs finden Sie unter [Verwalten von Desktopcomputerkonten](#).

Der Assistent zum Erstellen von Katalogen führt Sie durch alle Schritte. Die Maschinenkatalogerstellung ist der erste Schritt zur Bereitstellung von Desktops für Benutzer.

Die Schritte zum Erstellen von Maschinenkatalogen richten sich nach den Aufgaben, die die Benutzer ausführen sollen, den Geräten, auf denen die Desktops und Anwendungen bereitgestellt werden, oder der gewünschten Benutzererfahrung.

Die Hauptunterschiede ergeben sich aus der Wahl einer der folgenden Maschinenkatalogtypen:

- Serverbetriebssystem empfiehlt sich, wenn Benutzer spezifische Aufgaben ausführen und keine persönlichen Benutzereinstellungen für ihre Arbeit benötigen.
- Mit Desktopbetriebssystem können Sie sowohl jedem Benutzer eine eigene Desktopumgebung bereitstellen als auch benutzerdefinierbare Desktops einschließlich persönlicher vDisks (PvD).
- Über Remote-PC-Zugriff können Benutzer eine Verbindung mit ihren Bürocomputern remote über eine sichere

Verbindung herstellen.

Bevor Sie mit der Aufgabe Maschinenkatalog erstellen beginnen, stellen Sie sicher, dass alle erforderlichen Voraussetzungen für den jeweils gewünschten Katalogtyp gegeben sind.

Maschinenkatalogtyp	Anforderungen
Alle	• Einen Host mit genügend Prozessoren, Arbeitsspeicher und Speicher zur Unterbringung der Maschinen, die Sie erstellen möchten. • Wenn es mehrere Netzwerkkarten (NICs) gibt, müssen Sie entscheiden, welche virtuellen Netzwerke welcher Karte zugeordnet werden sollen.
Desktopbetriebssystem und Serverbetriebssystem	• Entweder eine ausreichende Anzahl an nicht verwendeten Active Directory-Computerkonten für die Maschinen, die Sie erstellen möchten, oder Zugriff auf ein Active Directory-Domänenadministratorkonto für die Domäne, in der die Desktops Mitglieder sein sollen. • Masterimage oder Vorlage zur Erstellung der Desktops. Das Masterimage muss auf dem Host verfügbar sein, auf dem die Maschinen erstellt werden.
Remote-PC-Zugriff	Geben Sie Maschinenkonten oder Organisationseinheiten an. Über diese Features können Sie die Maschinen nach Benutzern oder Benutzergruppen hinzufügen. Zum Aktivieren der Energieverwaltung für Maschinen des Remote-PC-Zugriff-Maschinenkatalogs wählen Sie die Option zur Verwendung einer Energieverwaltungsverbindung. Mit dieser Option können Benutzer ihre Maschinen remote einschalten.
Vorhandene	• VMs oder dedizierte Blade-PCs, auf denen Benutzerdesktops ausgeführt werden, die Sie bereits in das Datenzentrum migriert haben. • Benutzer- und Computerkonten in Active Directory zum Zuweisen zu den VMs oder Blade-PCs.
Anpassbarer Desktopbetriebssystem-Maschinenkatalog	• Eine Provisioning Services-Bereitstellung mit einer persönlichen vDisk, die Sie als Image vom Master-Zielgerät erstellt haben. • Gerätesammlungen, die konfiguriert wurden, die persönliche vDisk über das Netzwerk zu laden. • Von Provisioning Services verwaltete Active Directory-Computerkonten für jedes Zielgerät in der Gerätesammlung. • Wählen Sie die Art der Benutzererfahrung, die Sie bereitstellen möchten: • Verbindung mit einem zufälligen oder dedizierten Desktop bei der Anmeldung • Änderungen beibehalten oder verwerfen. Wenn Benutzer die Änderungen beibehalten, legen Sie fest, wo die Änderungen gespeichert werden sollen (auf separaten persönlichen vDisks oder auf einer lokalen Festplatte).
Serverbetriebssystem	Benutzer- und Computerkonten in Active Directory zum Zuweisen zu den VMs oder Blade-

Führen Sie zum Erstellen eines neuen Maschinenkatalogs die folgenden Aktionen aus:

1. Melden Sie sich bei dem Computer, auf dem Studio ausgeführt wird, an. Wenn Sie Studio zum Erstellen neuer Active Directory-Computerkonten für die Maschinen verwenden möchten, melden Sie sich mit einem Domänenadministratorkonto für die Domäne an, der Sie die Desktops hinzufügen möchten.
2. Wählen Sie den Knoten Maschinenkataloge und klicken Sie auf Maschinenkatalog erstellen.
Wenn der Knoten Maschinenkatalog nicht angezeigt wird, müssen Sie vor dem Erstellen des ersten Maschinenkatalogs eine der Erstkonfigurationsaufgaben abschließen, die beim Start von Studio angezeigt wurde.
3. Wählen Sie auf der Seite Betriebssystem und Hardware den gewünschten Typ des Maschinenkatalogs aus:
 - Desktopbetriebssystem bietet benutzerdefinierte und anpassbare Desktops.
 - Serverbetriebssystem bietet einen standardisierten Desktop.
 - Remote-PC-Zugriff bietet Benutzern den Remotezugriff auf ihre Bürocomputer.
4. Wählen Sie für Maschinenverwaltung den Typ der Maschine aus, über die die Desktops und Anwendungen bereitgestellt werden:
 - Virtuelle Maschinen: Verwenden Sie diesen Typ für per Cloud bereitgestellte Maschinen (einschließlich Citrix CloudPlatform und AWS).
 - Physische Hardware
5. Wählen Sie die Methode zur Maschinenimageverwaltung aus:
 - Maschinenerstellungsdienste (MCS)
 - Wenn keine Hostinginfrastruktur eingerichtet wurde oder Sie nicht darauf zugreifen können, werden die Optionen für die Verwendung von virtuellen Maschinen und Maschinenerstellungsdiensten deaktiviert. Informationen zum Einrichten der Hostinginfrastruktur finden Sie unter [Verbindungen und Ressourcen](#).
 - Zum Erstellen eines Maschinenkatalogs, der GPU-Ressourcen verwendet, können Sie Maschinenerstellungsdienste nur für die Erstellung eines GPU-fähigen Masterimages verwenden.
 - Provisioning Services: Wählen Sie den Provisioning Services-Server, indem Sie die IP-Adresse und die Domäne eingeben. Provisioning Services wird für per Cloud bereitgestellte Maschinen (einschließlich Citrix CloudPlatform und AWS) nicht unterstützt.
 - Anderer Dienst oder andere Technologie (vorhandene Images)
6. Legen Sie auf der Seite Desktoperfahrung (nicht verfügbar für Remote-PC-Zugriff) fest, wie sich Benutzer anmelden:
 - Legen Sie fest, wie viele Benutzer eine Verbindung zu derselben Maschine herstellen können.
 - Desktopbetriebssystemmaschinen:
 - Wählen Sie "Zufälliger Desktop" oder "Mit dem gleichen (statischen) Desktop".
 - Legen Sie fest, ob Änderungen beibehalten werden sollen, und wenn ja, wie sie gespeichert werden sollen: Personal vDisk oder dedizierte virtuelle Maschine.
 - Desktopbetriebssystemmaschinen mit Provisioning Services: Zufälliger Desktop oder derselbe statische Desktop.
7. Seite Maschinen:
 - Wählen Sie die Anzahl der für den Katalog zu erstellenden Maschinen aus.
 - Wählen Sie die Maschinenspezifikationen, z. B. CPU und Speicher, aus. Bei per Cloud bereitgestellten Maschinen ist dies ein CloudPlatform-Dienstangebot oder eine AWS-Instanz, die vom Cloudadministrator eingerichteten Hardwareprofilen für virtuelle Maschinen zugeordnet ist.
 - Fügen Sie Computerkonten hinzu:
 - Active Directory-Konten für Desktopbetriebssystem- und Serverbetriebssystem-Maschinenkataloge
 - Organisationseinheiten für Remote-PC-Zugriff-Maschinenkataloge
8. Klicken Sie auf der Seite Benutzer auf Einige der Maschinen führen Windows XP oder Windows Vista aus, wenn Sie

Maschinen im Maschinenkatalog haben, die unter Windows XP oder Vista laufen.

9. Wenn Sie eine Verwaltungsmethode für Maschinenimages gewählt haben, bei der ein Masterimage verwendet wird, wählen Sie auf der Seite Masterimage einen Host und den zugehörigen Snapshot bzw. die zugehörige virtuelle Maschine aus. Zum Erstellen eines Maschinenkatalogs mit GPU-Funktionen wählen Sie einen Snapshot oder ein Masterimage mit einer kompatiblen GPU-Ressource aus.
10. Wenn Sie eine Verwaltungsmethode gewählt haben, die virtuelle Maschinen erfordert, wählen Sie auf der Seite für das Hinzufügen und Konfigurieren virtueller Maschinen Folgendes:
 - Wählen Sie die Anzahl der virtuellen Maschinen.
 - Konfigurieren Sie die Anzahl der virtuellen CPUs sowie Speicher- und Festplattengröße
 - Geben Sie Größe und Laufwerksbuchstaben der persönlichen vDisk an.
11. Seite Netzwerkkarten:
 - Aktivieren Sie eine Netzwerkkarte, indem Sie eine vorhandene auswählen oder eine hinzufügen.
 - Wählen Sie das Netzwerk aus, das von der Netzwerkkarte verwendet wird.
12. Definieren Sie auf der Seite Geltungsbereiche, welche Administratoren auf den Maschinenkatalog zugreifen können.
13. Fügen Sie auf der Seite Zusammenfassung Beschreibungen hinzu und erstellen Sie den Maschinenkatalog.

Sie haben somit einen Maschinenkatalog mit den Maschinen für die Benutzerdesktops erstellt.

Um die Desktops der Maschinen des Katalogs für die Benutzer bereitzustellen, müssen Sie die Maschinen den Benutzern über Bereitstellungsgruppen zuweisen. Weitere Informationen finden Sie unter [Erstellen einer neuen Bereitstellungsgruppe](#).

Wenn Maschinenkataloge erstellt werden und das Masterimage mehrere Netzwerkkarten hat, konfigurieren Sie eine oder mehrere Karten für die Netzwerke. Beispiel: Wenn das Masterimage zwei Karten hat, können Sie eine der Karten für den Zugriff auf ein bestimmtes sicheres Netzwerk konfigurieren und die andere für den Zugriff auf ein mehr allgemeines Netzwerk einrichten.

1. Wählen Sie zum Erstellen von Katalogen im Assistenten den Knoten Konfiguration für Netzwerkkarten.
2. Wählen Sie eine Karte in der Anzeige und stellen Sie sicher, dass sie aktiviert ist.
3. Wählen Sie mindestens ein virtuelles Netzwerk zur Zuordnung für die ausgewählte Karte.
4. Fügen Sie optional Karten hinzu bzw. löschen Sie sie und klicken Sie dann auf Weiter, um die Maschinenkatalogerstellung fortzusetzen.

Zum Sicherstellen, dass Ihr Maschinenkatalog richtig für die Bereitstellung von Desktops oder Anwendungen konfiguriert ist, können Sie den Maschinenkatalog testen. In folgenden Fällen wird ein Maschinenkatalogkonfigurationstest ausgeführt:

- Automatisch nach dem Erstellen des Maschinenkatalogs
 - Durch Klicken auf Maschinenkatalog testen:
 - Auf der Seite Häufige Aufgaben
 - Auf der Seite Schnellstart
 - Als Aktion für einen Maschinenkatalog auf der Seite Maschinenkataloge
1. Klicken Sie zum Anzeigen der Konsole auf Studio.
 2. Wählen Sie einen Maschinenkatalog und klicken Sie auf Katalog testen.
- Wenn der Test abgeschlossen ist:
- Zeigen Sie optional einen Testbericht an
 - Beheben Sie die Probleme und testen Sie noch einmal
 - Richten Sie weitere Studio-Komponenten ein

Auswählen der Desktopumgebung

Sep 29, 2015

Der Maschinenkatalogtyp definiert die Hostinginfrastruktur für Desktops und Anwendungen und legt fest, wie viel Kontrolle die Benutzer über ihre Umgebung haben. Bei der Entscheidung über den Katalogtyp sollten Sie die Aufgaben berücksichtigen, die die Benutzer ausführen und die Geräte, auf denen die Desktops bereitgestellt werden.

Erstellen Sie Maschinenkataloge mit dem Assistenten zum Erstellen von Maschinenkatalogen. Sie müssen im Assistenten Angaben zu Ihrer Umgebung machen (nachfolgend beschrieben), mit denen Sie dann mit dem Assistenten einen für Ihre Anforderungen geeigneten Maschinenkatalogtyp auswählen können.

Wählen Sie einen Maschinenkatalog, der Desktops auf der Basis einer der folgenden Umgebungen bereitstellt:

Serverbetriebssystem

Serverbetriebssystem-Maschinenkataloge bieten eine Windows-Serverumgebung und Standarddesktops und Anwendungen, die gemeinsam von einer großen Anzahl von Benutzern verwendet werden.

Hinweis: Amazon Web Services (AWS) unterstützt nur Maschinenkataloge für Serverbetriebssystemmaschinen und [Server-VDI](#).

Diese Maschinenkataloge bieten Desktops und Anwendungen mit folgenden Merkmalen:

- Zuordnung zu Benutzern pro Sitzung nach dem Windhundverfahren
- Bereitstellung auf standardisierten Maschinen

Benutzer:

- Dies sind aufgabenorientierte Mitarbeiter, die standardisierte virtuelle Desktops und Anwendungen benötigen, z. B. Callcenter- und Einzelhandels-Arbeitskräfte.
- Sie müssen oder dürfen keine Anwendungen auf ihren Desktops installieren.

Ziel:

- Optimierung der verwendeten Hardware, indem Sie nicht jedem Benutzer einen bestimmten Desktop zuweisen, sondern nur so viele Desktops bereitstellen, wie zu einem beliebigen Zeitpunkt maximal erforderlich sind.
- Steuerung der Desktops und Erhöhen der Sicherheit, indem Sie die Benutzer daran hindern, dauerhafte Änderungen vorzunehmen.
- Minimieren der Desktopverwaltungskosten, indem Sie Ihren Benutzern eine gesperrte standardisierte Umgebung bereitstellen.

Desktopbetriebssystem

Desktopbetriebssystem-Maschinenkataloge bieten eine Windows-Desktopumgebung und Desktops und Anwendungen, die einzelnen Benutzern zugewiesen werden.

Benutzer:

- Sind aufgabenorientierte Mitarbeiter oder Fachkräfte, die personalisierte Desktops benötigen, die sie allein verwenden können.

- Sind mobile Mitarbeiter, die von verschiedenen Geräten aus in unterschiedlichen Netzwerken auf denselben Desktop zugreifen möchten.
- Möchten eigene Anwendungen auf ihren Desktops installieren.

Ziel:

- Bestimmte Aspekte der Benutzerdesktops über eine allgemeine Vorlage standardisieren.
- Benutzerdesktops auf allen Geräten bereitstellen, unabhängig von den Hardwaremerkmalen.
- Desktopverwaltungskosten reduzieren und den Benutzern gleichzeitig personalisierte Desktops bereitstellen.

Remote-PC-Zugriff

Remote-PC-Zugriff-Maschinenkataloge bieten Benutzern den Remotezugriff auf ihre physischen Bürodesktops. Benutzer sollen jederzeit an jedem Standort von jedem Gerät aus arbeiten können.

Hinweis: Es wird eine Benachrichtigung angezeigt, wenn Maschinen unter Microsoft Windows XP oder mit einer Virtual Delivery Agent-Version vor XenDesktop 7 ausgeführt werden. Diese Maschinen können zwar in separaten Maschinenkatalogen und Bereitstellungsgruppen eingeschlossen werden, können jedoch einige der neuen Features in diesem Release nicht verwenden.

Wählen Sie die Maschineninfrastruktur des Maschinenkatalogs. Die folgende Terminologie beschreibt die Optionen im Maschinenkatalog-Assistent, nicht die normalen Definitionen. Hierbei bezieht sich physisch nicht auf ein hardwarebasiertes Gerät und virtuell ist nicht unbedingt eine Softwareimplementierung einer Computerumgebung.

Verwenden von virtuellen Maschinen

Wählen Sie diesen Typ für Maschinenkataloge, die auf Maschinen basieren, bei denen die Energieverwaltung über Studio erfolgt, bzw. die über eine Cloudumgebung bereitgestellt werden.

Hinweis: Wenn beim Erstellen des Maschinenkatalogs keine Hypervisor-Verbindungen konfiguriert werden, ist diese Option nicht verfügbar.

Verwenden von physischer Hardware

Wählen Sie diesen Typ für Maschinenkataloge, die auf Maschinen basieren, deren Energieverwaltung nicht über Studio erfolgt.

Die Art der Verwaltung und der Maschinenerstellung wirkt sich auf den empfohlenen Maschinenkatalogtyp aus. Sie verwalten Maschinenimages eines Maschinenkatalogs mit einer der folgenden Methoden:

Die **Maschinenerstellungsdienste** verwalten virtuelle Maschinen in Ihrer Umgebung mit einer Master-VM, sodass Sie Zielgeräte über ein Masterimage verwalten und aktualisieren können.

Provisioning Services verwalten Zielgeräte als Gerätesammlung. Die Desktops und Anwendungen werden über eine Provisioning Services-vDisk bereitgestellt, deren Image von einem Masterzielgerät erstellt wurde. So können Sie die Rechenleistung der vorhandenen Hardware oder virtuellen Maschinen nutzen.

Hinweis: Die Verwendung von Provisioning Services wird zusammen mit AWS und CloudPlatform nicht unterstützt.

Mit **vorhandenen Images** (aus einem anderen Dienst bzw. einer anderen Technologie) werden die Desktops und Anwendungen verwaltet und bereitgestellt, die bereits auf VMs im Datacenter migriert wurden. Sie müssen die Zielgeräte individuell oder kollektiv mit ESD-Tools (Electronic Software Distribution) verwalten.

Wählen Sie den Typ des Desktops, der Benutzern bei der Anmeldung zugewiesen wird.

Wählen Sie eine der folgenden Optionen:

- Wie viele Benutzer können eine Verbindung zu derselben Maschine herstellen? – Legen Sie einen oder mehrere fest.
- Benutzer sollen bei jeder Anmeldung mit einem neuen (zufälligen) Desktop verbunden werden: Es werden keine Benutzeränderungen gespeichert.
- Benutzer sollen bei jeder Anmeldung mit dem gleichen (statischen) Desktop verbunden werden: Wählen Sie eine der folgenden Optionen für das Speichern von Änderungen aus:
 - Änderungen auf separater persönlicher vDisk speichern
 - Dedizierte VM erstellen und Änderungen auf lokalem Datenträger speichern Bei dieser Auswahl müssen Sie folgende Informationen zum Speichern der Benutzerdaten eingeben:
 - Größe der persönlichen vDisk
 - Laufwerksbuchstabe
 - Änderungen bei Abmeldung verwerfen und virtuelle Desktops bei Abmeldung entfernen

Anzahl und Rechenleistung

Wählen Sie die Anzahl der für den Katalog zu erstellenden virtuellen Maschinen aus.

Wählen Sie Spezifikationen für die virtuellen Maschinen aus. Wählen Sie für virtuelle Maschinen im lokalen Datacenter Einstellungen für RAM und CPU aus.

Bei einer Cloudumgebung wird die Rechenleistung virtueller Maschinen (z. B. RAM- und CPU-Einstellungen) wie folgt genannt:

- Dienstangebot: CloudPlatform
- Instanztyp: AWS

Dienstangebote und Instanztypen werden als einzelner Wert festgelegt, häufig als **Klein**, **Mittel** und **Groß** festgelegt. Die Zuordnung dieser Werte zu den Hardwareprofilen virtueller Maschine wird vom Cloudanbieter, nicht Studio, festgelegt.

Maschinenvorlagen

Wählen Sie für AWS-Umgebungen die Vorlage für virtuelle Maschinen aus, mit der Sie Zielgeräte einfach verwalten und aktualisieren können.

Vorbereiten eines Masterimages

Sep 29, 2015

Zur Bereitstellung von Desktops und Anwendungen für die Maschinen von Serverbetriebssystem- oder Desktopbetriebssystem-Maschinenkatalogen müssen Sie das Masterimage vorbereiten, mit dem die Benutzerdesktops und Anwendungen erstellt werden.

Das Masterimage ist eine Vorlage, die Sie für die Bereitstellung der Umgebung verwenden. Zusätzlich zu den Desktops und Anwendungen werden beim Erstellen des Masterimage das Betriebssystem und die Software installiert und konfiguriert.

Beachten Sie Folgendes:

- Remote-PC-Zugriff-Maschinenkataloge benötigen keine Masterimages.
- Bei Cloudbereitstellungen werden Vorlagen statt Masterimages verwendet. Beachten Sie bei der Vorbereitung von Vorlagen folgende Informationen:
 - Bei Verwendung von Amazon Web Services (AWS) lesen Sie den Abschnitt *— Erstellen eines XenApp- oder XenDesktop-Infrastrukturstacks mit der CloudFormation-Vorlage* in [Bereitstellen von XenApp 7.5 und XenDesktop 7.5 mit Amazon VPC](#).
 - Bei Verwendung von Citrix CloudPlatform lesen Sie den Abschnitt *— Working with Templates* im Handbuch [CloudPlatform Version 4.2 Administrator's Guide](#).

Wichtig: Wenn Sie Provisioning Services oder Maschinenerstellungsdienste verwenden, führen Sie auf den Masterimages nicht Sysprep aus.

Microsoft KMS-Aktivierung mit Maschinenerstellungsdiensten

Wenn Sie einen XenDesktop 7.x- oder XenApp 7.5-VDA mit XenServer 6.1/6.2, vSphere oder Microsoft System Center Virtual Machine Manager (VMM) verwenden, müssen Sie Microsoft Windows oder Microsoft Office nicht manuell zurücksetzen, wenn Sie die Maschinenerstellungsdienste verwenden.

Wenn Sie einen XenDesktop 5.x-VDA mit XenServer 6.0.2 verwenden, führen Sie bei Verwendung der Maschinenerstellungsdienste die Schritte durch, die unter [CTX128580](#) beschrieben sind.

Windows Desktop Experience mit Serverbetriebssystemmaschinen

Sie müssen das Windows Desktop Experience Feature Pack einem Masterimage für Serverbetriebssystemmaschinen hinzufügen, das unter folgenden Betriebssystemen ausgeführt wird:

- Windows Server 2008 R2
- Windows 7 N Edition
- Windows XP

Ohne dieses Feature Pack gibt Windows Media Player Video in einer Sitzung nicht richtig wieder. Folgen Sie den Schritten, die in [CTX133429](#) aufgeführt sind.

GPU-Masterimages

Bei GPU-fähigen Maschinen und Maschinenkatalogen müssen Sie ein GPU-dediziertes Masterimage verwenden. Virtuelle Maschinen mit GPU benötigen Videotreiber, die GPUs unterstützen und so konfiguriert sind, dass die VM Software verwenden kann, die den GPU für Operationen verwendet. Sie können das Masterimage mit der XenCenter-Konsole von

XenServer erstellen. Genaue Informationen finden Sie in der XenServer-Dokumentation.

Im Folgenden werden die allgemeinen Schritte zum Erstellen eines GPU-Masterimages beschrieben, das von Studio erkannt wird, wenn Sie GPU-fähige Ressourcen, wie unter [Verbindungen und Ressourcen](#) beschrieben, auswählen.

1. Erstellen Sie mit XenCenter eine VM mit Standard-VGA sowie Netzwerken und einer vCPU.
2. Aktualisieren Sie die VM-Konfiguration um anzugeben, dass der GPU (entweder Passthrough oder vGPU) verwendet werden kann.
3. Installieren Sie ein unterstütztes Betriebssystem auf der VM.
4. Installieren Sie die XenServer-Tools auf der VM.
5. Aktivieren Sie RDP innerhalb des Betriebssystems der VM.
6. Installieren Sie NVIDIA-Treiber.
7. Deaktivieren Sie die VNC-Verwaltungskontrolle (Virtual Network Computing) über XenCenter, um die Leistung zu optimieren, und starten Sie anschließend die VM neu.
Sie werden aufgefordert, RDP zu verwenden.
8. Verwenden Sie RDP zum Installieren der VDA-Software und führen Sie einen Neustart durch.
9. Optional können Sie einen Snapshot von der VM erstellen und als Vorlage für andere GPU-Masterimages verwenden.
10. Verwenden Sie RDP zum Installieren kundenspezifischer Anwendungen, die GPU-Funktionen umfassen und in XenCenter konfiguriert wurden.

Vorbereiten eines Masterimages

1. Erstellen Sie die Serverbetriebssystem- oder Desktopbetriebssystem-Maschinenkataloge mit dem Verwaltungstool für den Hypervisor zum Erstellen eines neuen Masterimage und Installieren des Betriebssystems (einschließlich aller Service Packs und Updates).

Wenn die Anzahl der vCPUs und der Arbeitsspeicherumfang zur Ausführung des Masterimage ausreichen, sind diese dem Masterimage zugewiesenen Faktoren in diesem Stadium nicht kritisch, da Sie diese Einstellungen beim Erstellen des Maschinenkatalogs ändern können.

Wichtig: Stellen Sie sicher, dass Sie das Masterimage mit der gleichen Festplattenspeicherplatzgröße einrichten, die für die Benutzerdesktops und Anwendungen erforderlich ist, da dieser Wert später nicht geändert werden kann.

Stellen Sie sicher, dass die Festplatte für das Masterimage am Gerätestandort 0 verbunden ist. Dieser Standort in den meisten Standardmasterimagevorlagen automatisch konfiguriert; in einigen benutzerdefinierten Vorlagen kann dies nicht der Fall sein.

Für über Provisioning Services (PVS) bereitgestellte Maschinen können Sie entweder ein Masterimage oder einen physikalischen Computer als Master-Zielgerät verwenden.

2. Installieren Sie die jeweiligen Integrationstools für Ihren Hypervisor (XenServer Tools, Hyper-V Integration Services oder VMware Tools) auf dem Masterimage.
Hinweis: Wenn Sie die Hypervisor-Integrationstools nicht auf dem Masterimage installieren, funktionieren Ihre Desktops möglicherweise nicht richtig.
3. Installieren und konfigurieren Sie Virtual Delivery Agent (VDA) vom Installationsmedium wie unter [Installieren über die grafische Oberfläche](#) beschrieben. Bei der Installation von VDA wählen Sie die Option zur Desktopoptimierung. Hierdurch wird die Leistung der Desktops und Anwendungen der Benutzer verbessert, indem verschiedene Windows-Funktionen neu konfiguriert werden, die mit virtuellen Desktops nicht kompatibel sind oder nicht benötigt werden.
4. Installieren Sie Tools von Drittanbietern, die Sie ausführen möchten, wie zum Beispiel Antivirensoftware oder elektronische Softwarebereitstellungsagenten, und konfigurieren Sie Dienste, die für Ihre Bereitstellung erforderlich sind,

wie zum Beispiel Windows Update. Stellen Sie sicher, dass Sie die entsprechenden Einstellungen für Ihre Benutzer und den Maschinentyp verwenden, da diese Konfigurationen über das Masterimage auf Benutzerdesktops und Anwendungen übertragen werden.

5. Installieren und konfigurieren Sie Anwendungen von Drittanbietern, die Sie nicht virtualisieren möchten. Citrix empfiehlt, dass Sie Anwendungen virtualisieren. Dadurch können Sie die Desktopverwaltungskosten erheblich reduzieren, da das Masterimage nicht jedes Mal aktualisiert werden muss, wenn Sie eine Anwendung hinzufügen oder neu konfigurieren möchten. Je weniger Anwendungen auf den einzelnen Desktops installiert sind, desto weniger Speicherkosten fallen an, da die Größe der Masterimage-Festplatten reduziert werden kann.
6. Wenn Sie planen, App-V-Anwendungen über Studio zu veröffentlichen, wie unter [Verwalten der die App-V-Anwendungsbereitstellung](#) beschrieben, installieren und konfigurieren Sie App-V-Clients mit allen empfohlen App-V-Einstellungen auf dem Masterimage.
7. Bei über Maschinenerstellungsdienste (MCS) erstellten Maschinenkatalogen müssen Sie die Gebietsschemas und Sprachpakete auf dem Masterimage vor der Bereitstellung installieren, wenn eine Lokalisierung von Microsoft Windows nötig ist. Wenn ein Masterimage-Snapshot beim Provisioning erstellt wird, verwenden die bereitgestellten VMs die installierten Gebietsschemas und Sprachpakete.
8. Zum Bereitstellen von Serverbetriebssystem- oder Desktopbetriebssystemmaschinen desktops binden Sie das Masterimage in die Domäne ein, zu der die Benutzerdesktops und Anwendungen gehören sollen, und stellen das Masterimage auf dem Host zur Verfügung, auf dem Sie die Maschinen erstellen möchten.
Citrix empfiehlt, dass Sie einen Snapshot des Masterimages erstellen und diesen so benennen, dass Sie das Masterimage zukünftig gut daran identifizieren können. Wenn Sie ein Masterimage anstelle eines Snapshots beim Erstellen eines Serverbetriebssystem- oder Desktopbetriebssystem-Maschinenkatalogs angeben, erstellt Studio automatisch einen Snapshot, der jedoch nicht umbenannt werden kann.

Bei über Provisioning Services erstellten Maschinenkatalogen, wie unter [Erstellen eines neuen Maschinenkatalogs](#) beschrieben, erstellen Sie eine VHD-Datei für die vDisk von Ihrem Master-Zielgerät, bevor Sie das Master-Zielgerät in eine Domäne einbinden. Weitere Informationen zum Imaging für vDisk finden Sie unter [Installieren und Konfigurieren von Provisioning Services](#).

Bereitstellungsgruppen

Sep 29, 2015

Bereitstellungsgruppen sind Sammlungen von Maschinen, die angeben, welche Benutzer eine Gruppe von Desktops oder Anwendungen verwenden können. Erstellen Sie Bereitstellungsgruppen beispielsweise für bestimmte Teams, Abteilungen oder Benutzertypen. Bereitstellungsgruppen haben folgende Funktion:

- Angeben von Benutzergruppen für den Zugriff auf Desktops, Anwendungen oder Desktops und Anwendungen
- Hinzufügen von Benutzer und Benutzergruppen

Für die Bereitstellung von Anwendungen und Desktops fügen Sie Benutzer oder Benutzergruppen den Bereitstellungsgruppen hinzu. Sie können Benutzergruppen mit "Durchsuchen" auswählen oder eine Liste von Active Directory-Benutzern und -Gruppen eingeben.

Bei Desktopbetriebssystem-Bereitstellungsgruppen können Sie nach der Erstellung der Gruppe Benutzerdaten aus einer Datei importieren.

Die Eigenschaften einer Bereitstellungsgruppe basieren auf den Desktops oder Maschinen innerhalb des Maschinenkatalogs der Gruppe. Ein Desktop ist eine virtuelle oder physische Maschine, die manuell mit dem Assistenten Maschinenkatalog erstellen erstellt wurde. Bei Serverbetriebssystem- und Desktopbetriebssystem-Maschinenkatalogen kann die Bereitstellung auch automatisch mit den Maschinenerstellungsdiensten oder Provisioning Services erfolgen. Die Merkmale der Desktops stammen vom Maschinenkatalog, denen sie zugewiesen sind.

Ein Maschinenkatalog ist eine Sammlung physischer Computer und virtueller Maschinen, die Sie Benutzern über die Bereitstellungsgruppe zuweisen. Was der Benutzer sieht, hängt vom Maschinentyp ab:

- Serverbetriebssystemdesktops und -anwendungen: Die Benutzererfahrung entspricht einer Windows Server-Umgebung. Bei diesem Typ verwenden mehrere Benutzersitzungen eine Windows Server-Umgebung. Diese Option ist für Benutzer geeignet, die definierte Aufgaben ausführen und keine Personalisierung benötigen. Dies ist eine zufällige Desktopumgebung, in der Benutzer eine Verbindung mit einer verfügbaren Windows Server-Umgebung herstellen.
- Desktopbetriebssystemdesktops und -anwendungen: Die Benutzererfahrung entspricht einer Windows-Clientumgebung. Bei diesem Typ melden sich mehrere Benutzer an einer Windows-Clientumgebung an. Diese Option stellt die folgenden Desktopumgebungen bereit:
 - Statische Umgebung für Benutzer; die sich mit einem personalisierten Desktop oder einer personalisierten Anwendung verbinden
 - Zufällige Umgebung, in der der Maschinenzustand bei der Benutzerabmeldung zurückgesetzt wird, die Maschine steht dann anderen Benutzern zur Verfügung.
- Remote-PC-Zugriff-Maschinen bieten Benutzern den Remotezugriff auf ihre Bürodesktops.

Weder Maschinenkataloge noch Bereitstellungsgruppen können Mischformen enthalten; sie müssen entweder nur aus Serverbetriebssystemmaschinen, nur aus Desktopbetriebssystemmaschinen oder nur aus Remote-PC-Zugriff-Maschinen bestehen.

Abhängig von der Umgebung, wie unter

— *Benutzererfahrung bei Bereitstellungsgruppen*

beschrieben, können Sie den Typ der Bereitstellungsgruppen auswählen, der festlegt, was Bereitstellungsgruppen den Benutzergeräten bereitstellen:

- nur Desktops
- nur Anwendungen
- Desktops und Anwendungen (bei statischen Desktopbetriebssystemmaschinen nicht verfügbar)

Das Beispielszenario dieses Abschnitts beschreibt die wichtigsten Schritte, die beim Einrichten zweier Bereitstellungsgruppen zu beachten sind. Eine der Bereitstellungsgruppen basiert auf einem Windows Server 2008 R2-Masterimage, entspricht in Aussehen und Verhalten aber einer Windows 7-Umgebung. Die andere Gruppe basiert auf einem Windows 8-Masterimage. Die folgenden Funktionen zur Personalisierung gewährleisten eine konsistente Benutzererfahrung bei der Anmeldung und anpassbare Anwendungen für Benutzer in den Bereitstellungsgruppen:

- Richtlinien und Active Directory-Gruppenrichtlinienobjekte ermöglichen die globale Verwaltung der Benutzer-Personalisierungseinstellungen über beide Bereitstellungsgruppen hinweg.
Bei einfachen Szenarios bietet es sich an, die Citrix Profilverwaltung zu verwenden, die automatisch im Rahmen der Installation von Virtual Delivery Agent auf Masterimages installiert wird. Sie können damit das Profilverhalten bereitstellungsgruppenorientiert definieren.
- Das Personal vDisks-Feature bietet weiterhin die Einzelimageverwaltung für gepoolte Desktops während die Benutzer gleichzeitig Anwendungen installieren und ihre Desktopeinstellungen anpassen können. Dieses Feature wird für die Bereitstellungsgruppe verwendet, die auf einem Windows 8-Image basiert.

Desktopstrategie

Eine neue Unternehmensdesktopstrategie bedeutet, dass Sie so viele Desktops wie möglich virtualisieren sollten. Bei diesem Beispiel werden zunächst die Desktops von zwei Teams einbezogen.

- Das Buchhaltungsteam verwendet viele physische Arbeitsstationen, auf denen mehrere Buchhaltungsstandardanwendungen installiert sind. Auf den Arbeitsstationen wird Windows 7 ausgeführt. Um diese durch virtuelle Desktops zu ersetzen, erstellen Sie eine Windows Server 2008 R2-Bereitstellungsgruppe mit Windows 7-Aussehen und -Verhalten für dieses Team.
Mitglieder des Buchhaltungsteams müssen keine eigenen Anwendungen installieren, daher wird Personal vDisks nicht benötigt. Allerdings passen die Mitarbeiter normalerweise Anwendungen auf ihren physischen Maschinen an eigene Anforderungen an (z. B. Auswahl einer Homepage). Um die personalisierten Einstellungen in der neuen virtualisierten Umgebung beizubehalten, verwenden Sie die Profilverwaltungsrichtlinien.
- Das Vertriebsteam möchte ein Upgrade der Arbeitsstationen, die zum Testen von Kundenanwendungen eingesetzt werden, auf Windows 8 durchführen. Das Team muss die Desktops selbst installieren und deinstallieren können. Verwenden Sie daher Personal vDisks, um dies zu ermöglichen.

Desktopumgebung

Bei diesem Szenario wird angenommen, dass Ihre Umgebung folgende Voraussetzungen erfüllt:

- Das Netzwerk hat schnelle Verbindungen mit geringer Latenz.
- Der Schätzwert für den erforderlichen Speicherplatz für Personal vDisks basiert auf dem tatsächlichen und erwarteten Personalisierungsniveau im Unternehmen.

Verwenden Sie zum effektiven Implementieren der neuen Desktopstrategie folgende erforderlichen Komponenten:

- Unterstützter Hypervisor
- XenApp oder XenDesktop
- Bereitstellungslösung:
 - Maschinenerstellungsdienste (MCS) sind standardmäßig verfügbar, wenn beim Erstellen einer Anwendungs- und Desktopbereitstellungssite ein Host konfiguriert wird und festgelegt wird, dass zum Erstellen von VMs MCS verwendet wird.
Die Maschinenerstellungsdienste bieten eine konsolidierte Verwaltung über Citrix Studio.
 - Citrix Provisioning Services (PVS) ist eine weitere Option für die Verwendung mit System Center 2012 Virtual Machine Manager. Zur Vorbereitung für Provisioning Services müssen Sie den Provisioning Services-Server separat installieren und ein Provisioning Services-vDisk-Masterimage konfigurieren. Bei der VDA-Installation wird der Provisioning Services-Agent auf den Masterimages für Windows 7 und Windows 8 installiert.
Sie müssen die Provisioning Services Console zum Verwalten von Provisioning Services verwenden.

Hinweis: Zum Bereitstellen nicht verwalteter physischer Desktops ist keine Bereitstellungslösung erforderlich.

- Masterimage für Windows Server 2008 R2 und Windows 8
- Citrix Receiver (installiert auf einem Benutzergerät) zum Testen des Zugriffs auf die neuen virtuellen Desktops

Studio

Studio ist die Verwaltungskonsolle, mit der Sie die Bereitstellung konfigurieren und verwalten, sodass für die Bereitstellung von Anwendungen und Desktops keine separaten Verwaltungskonsolen benötigt werden.

Studio ist ein Snap-In für Microsoft Management Console, das alle Objekte Ihrer Bereitstellung anzeigt. Es bietet diverse Assistenten, die Ihnen bei der Einrichtung Ihrer Umgebung und der Arbeitslast zum Hosten von Anwendungen und Desktops sowie beim Zuweisen von Anwendungen und Desktops für Benutzer behilflich sind.

Konfiguration

Nach der Installation der erforderlichen Komponenten konfigurieren Sie die Umgebung mit den folgenden allgemeinen Schritten. Ausführliche Anweisungen finden Sie unter [Erstellen einer Site](#).

1. Um die Kontinuität der Betriebsabläufe sicherzustellen, aktivieren Sie die Funktion "AlwaysOn-Verfügbarkeitsgruppen" in SQL Server 2012.
2. Nachdem Sie die Spiegeldatenbanksoftware auf einem anderen Server installiert haben, verwenden Sie Studio zum Erstellen einer vollständigen Bereitstellungssite auf dem Server, auf dem Sie Delivery Controller installiert haben. Abhängig von den Berechtigungen für die Datenbank konfiguriert Studio die Haupt- und die Spiegeldatenbank der Sitekonfiguration, oder es können Skripts generiert werden, die vom Datenbankadministrator zum Konfigurieren benutzt werden. Der Controller erkennt dann automatisch, dass die Sitekonfigurationsdatenbank repliziert wurde und verwendet ggf. die Spiegelung.
Wählen Sie beim Erstellen der Site den Speicherort für persönliche vDisks. Obwohl persönliche vDisks (VHD-Datenträger) physisch auf dem Hypervisor gespeichert werden, müssen sie nicht an demselben Speicherort sein wie andere, dem virtuellen Desktop angeschlossene Datenträger. Dadurch können sich die Kosten für Personal vDisk-Speicher verringern.
3. Verwenden Sie den Assistenten zum Erstellen von Maschinenkatalogen, um zwei Kataloge für die zwei Masterimages zu erstellen:
 - Einen Serverbetriebssystem-Maschinenkatalog für die Windows Server 2008 R2-Desktops des Buchhaltungsteams

Wenn Sie Maschinenerstellungsdienste verwenden: Da Benutzer nicht bei jeder Anmeldung eine Verbindung mit derselben Instanz eines Desktops herstellen müssen, legen Sie für virtuelle Maschinen den Typ "Zufällig" fest.

- Einen Desktopbetriebssystem-Maschinenkatalog für die Windows 8-Desktops des Vertriebsteams
Wenn Sie mit Provisioning Services arbeiten: Erstellen Sie den Katalog, indem Sie eine Verbindung mit einem Provisioning Services-Server herstellen und eine geeignete Gerätesammlung auswählen. Da die Benutzer sehr stark personalisierte Desktops benötigen, legen Sie virtuelle Maschinen des Typs "Statisch" fest und verbinden Sie sie mit persönlichen vDisks. Geben Sie auch die Größe der vDisks an und den für sie zu verwendenden Laufwerksbuchstaben.

Tipp: Denken Sie daran, den Personal vDisk-Bestand auszuführen, wenn Sie das Masterimage des Desktopbetriebssystem-Maschinenkatalogs ändern. Wann dies geschieht, hängt von der jeweiligen Provisioning Services-Einrichtung ab. Weitere Informationen finden Sie unter [Provisioning Services](#).

4. Erstellen Sie mit dem Assistenten zum Erstellen von Bereitstellungsgruppen zwei Bereitstellungsgruppen für Ihre beiden Kataloge:

- Eine, die Desktops und Anwendungen für das Buchhaltungsteam enthält
- Eine, die Desktops für das Vertriebsteam enthält

Fügen Sie jeder Bereitstellungsgruppe Konten für die Benutzer des Teams hinzu (sodass die richtigen Personen auf die richtigen Desktops zugreifen können). Fügen Sie außerdem ein Testbenutzerkonto für jedes Team hinzu.

Um sicherzustellen, dass die Benutzer des Buchhaltungsteams auch an einem anderen, zufällig zugewiesenen Desktop dieses Teamkatalogs wie gewohnt arbeiten können, verwenden Sie Profildefinitionsrichtlinien zum Festlegen der von Features, z. B. des Speicherorts von umgeleiteten Ordnern.

Testen

Testen Sie die Benutzererfahrung wie folgt:

1. Vergewissern Sie sich, dass die neuen Desktops beim Controller registriert sind, indem Sie Ihre Bereitstellungsgruppen in Studio auswählen und auf Bereitstellungsgruppe testen klicken. Folgen Sie den Empfehlungen für alle eventuell angezeigten Fehler.
2. Melden Sie sich auf dem Benutzergerät bei Receiver als Mitglied des Buchhaltungsteams an.
Wird der Windows 7-Desktop richtig in Desktop Viewer angezeigt? Entspricht er in Aussehen und Verhalten dem Windows 7-Desktop, obwohl er über eine Windows Server-Maschine erstellt wurde? Ändern Sie eine Anwendungseinstellung oder eine Voreinstellung, die in Ihrem Profil gespeichert wird, und melden Sie sich anschließend ab und erneut an. Wurde die neue Anwendungseinstellung oder Voreinstellung gespeichert?
3. Wiederholen Sie den letzten Schritt unter Anmeldung bei Receiver als Mitglied des Vertriebsteams.
Wird der Windows 8-Desktop richtig in Desktop Viewer angezeigt? Haben Sie ein Laufwerk (persönliche vDisk) mit dem beim Erstellen des Maschinenkatalogs für diesen Desktop angegebenen Laufwerksbuchstaben? Können Sie auf diesem Laufwerk Anwendungen installieren und Daten speichern?

Bereitstellungsgruppen geben an, welche Benutzer Zugriff auf Desktops oder Anwendungen haben, und zwar normalerweise auf der Basis von Benutzermerkmalen wie z. B. dem Benutzertyp. Vor dem Erstellen einer Bereitstellungsgruppe, beachten Sie Folgendes:

- Sie können eine Bereitstellungsgruppe nur erstellen, wenn mindestens eine Maschine in dem ausgewählten Katalog ungenutzt bleibt.
- Sie können eine Maschine nur in einer einzigen Bereitstellungsgruppe verwenden.

- Bereitstellungsgruppen können mit mehreren Maschinenkatalogen mit den gleichen Merkmalen erstellt werden.
- Ein Maschinenkatalog kann einer oder mehreren Bereitstellungsgruppen zugeordnet werden.
- Mehrere Kataloge können auf dieselbe Bereitstellungsgruppe verweisen.
- In Studio können Sie keine gemischten Bereitstellungsgruppen mit Maschinenkatalogen mit unterschiedlichen Maschinentypen erstellen. Wenn Sie die Maschinen in eine einzelne Gruppe aufnehmen, müssen die Katalogmerkmale übereinstimmen. Sie können beispielsweise Maschinen nicht wie folgt mischen:
 - Serverbetriebssystem-Maschinenkataloge mit Desktopbetriebssystem-Maschinenkatalogen
 - Maschinenkataloge mit statischer Zuweisung mit Maschinenkatalogen mit zufälliger Zuweisung
- Bei Remote-PC-Zugriff-Bereitstellungsgruppen werden alle im Remote-PC-Maschinenkatalog hinzugefügten Maschinen automatisch einer Bereitstellungsgruppe zugeordnet.

Erstellen einer Bereitstellungsgruppe

1. Wählen Sie in Studio den Knoten Bereitstellungsgruppe und klicken Sie auf Bereitstellungsgruppe erstellen.
2. Klicken Sie auf Maschinen hinzufügen, wählen Sie einen Katalog für diese Bereitstellungsgruppe und geben Sie die Anzahl an Maschinen ein, die für die Gruppe aus dem Katalog verwendet werden sollen.
Tipp: Die Gesamtanzahl der verfügbaren Maschinen im Katalog wird angezeigt.
3. Klicken Sie auf der Seite Benutzer auf Benutzer hinzufügen, um die Benutzer oder Benutzergruppen hinzuzufügen, die auf die Desktops oder Anwendungen zugreifen können. Sie können Benutzergruppen auswählen, indem Sie die Liste der Active Directory-Benutzer und -Gruppen durchsuchen oder Benutzer und Benutzergruppen eingeben. Trennen Sie mehrere Einträge durch Semikolons. Bei Desktopbetriebssystem-Bereitstellungsgruppen können Sie nach der Erstellung der Gruppe Benutzerdaten aus einer Datei importieren.
4. Klicken Sie auf der Seite Bereitstellungstyp auf die Elemente, die den Benutzern bereitgestellt werden sollen:
 - nur Desktops
 - nur Anwendungen
 - Desktops und Anwendungen (bei statischen Desktopbetriebssystemmaschinen nicht verfügbar)
5. Wenn die Bereitstellungsgruppe Anwendungen bereitstellt, wählen Sie auf der Seite Anwendungen die gewünschten Anwendungen aus oder fügen Sie Anwendungen wie unter [Erstellen einer Bereitstellungsgruppenanwendung](#) beschrieben hinzu.
6. Klicken Sie auf der Seite StoreFront auf die StoreFront-URLs für Citrix Receiver, damit Receiver und StoreFront ohne Benutzereingriff verbunden werden können. Hinweis: Diese Einstellung gilt, wenn Receiver auf VDAs ausgeführt wird.
 - Wählen Sie Automatisch aus, um Server-URLs aus der Liste auszuwählen.
 - Wählen Sie Manuell aus, um eine Serveradresse einzugeben, die nicht in der Liste angezeigt wird.
 - Wählen Sie Nein aus, um keine Auswahl für StoreFronts zu treffen und gehen Sie zur Seite "Zusammenfassung".
 - Wählen Sie Ja, wenn Sie vorhandene URLs auswählen möchten.
Enthält die Anzeige keine passenden StoreFronts, können Sie auf Hinzufügen klicken und die URL eines existierenden Stores eingeben.
7. Definieren Sie auf der Seite Geltungsbereiche, welche Administratoren auf die Bereitstellungsgruppe zugreifen können.
8. Überprüfen Sie die Angaben auf der Seite Zusammenfassung und geben Sie einen Anzeigenamen ein, der für Benutzer und Administratoren sichtbar ist, sowie einen aussagekräftigen Bereitstellungsgruppennamen, der nur für Administratoren sichtbar ist.

Sie haben somit eine Bereitstellungsgruppe erstellt. Informationen über die Anwendungen, die von Bereitstellungsgruppen für Benutzerendgeräte verfügbar gemacht werden können, finden Sie unter [Gehostete Anwendungen](#).

In folgenden Fällen wird ein Bereitstellungsgruppenkonfigurationstest ausgeführt:

- Automatisch nach dem Erstellen einer Bereitstellungsgruppe
- Durch Klicken auf Bereitstellungsgruppe testen:
 - Auf der Seite Häufige Aufgaben
 - Als Aktion für eine Bereitstellungsgruppe auf der Seite Bereitstellungsgruppen

Testen einer Bereitstellungsgruppe

1. Klicken Sie zum Anzeigen der Konsole auf Studio.
2. Wählen Sie eine Bereitstellungsgruppe und klicken Sie auf Bereitstellungsgruppe testen.
3. Wenn der Test abgeschlossen ist, führen Sie die folgenden Aktionen aus:
 - Konfigurieren Sie die App-V-Ressourcen in Studio.
 - Fügen Sie App-V-Anwendungen den Bereitstellungsgruppen wie unter [Erstellen einer Bereitstellungsgruppenanwendung](#) beschrieben hinzu.
 - Verwalten Sie die Anwendungen wie unter [Ändern und Verwalten von Anwendungen](#) beschrieben.

Gehostete Anwendungen

Sep 29, 2015

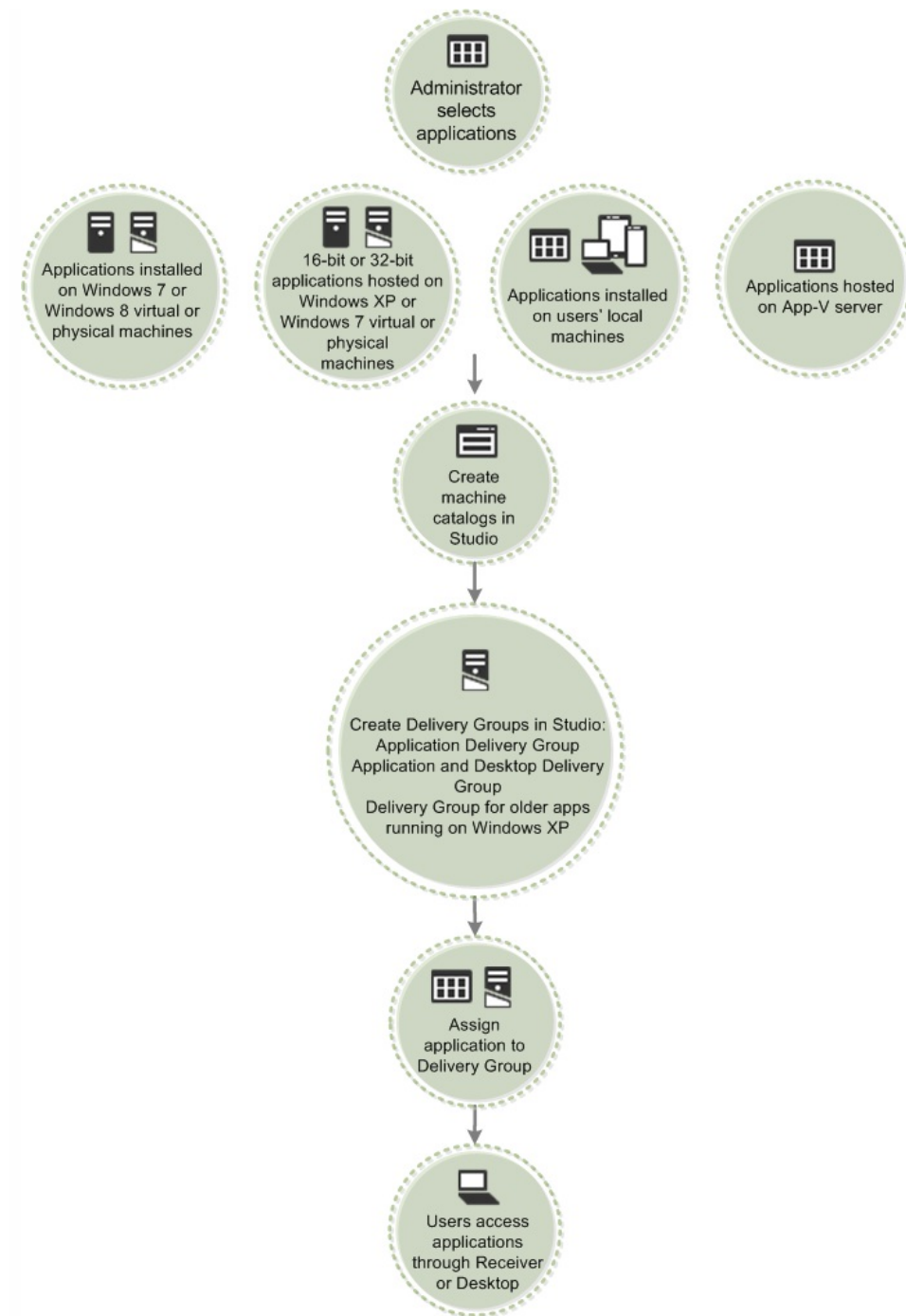
In Studio können Sie Anwendungen den Benutzergeräten mit verschiedenen Bereitstellungsmethoden bereitstellen.

Mögliche Bereitstellungsmethoden:

- Im Datacenter gehostete Anwendungen wie unter [Erstellen einer Bereitstellungsgruppenanwendung](#). Diese Anwendungen:
 - Werden auf physischen oder virtuellen Maschinen installiert und ausgeführt.
 - Gehören zu einem Masterimage, mit dem die Benutzerdesktops erstellt werden. Das Masterimage enthält die Elemente, die für alle Benutzer gelten, wie zum Beispiel Antivirensoftware, Citrix Plug-Ins und andere Standardprogramme.
- Lokaler App-Zugriff (LAA): Anwendungen, auf die zugegriffen wird, indem Verknüpfungen auf lokal installierte Anwendungen auf einem virtuellen Desktop veröffentlicht werden, wie unter [Lokaler App-Zugriff](#) beschrieben.
- Microsoft Application Virtualization-Anwendungen (App-V): Stellt App-V-Anwendungen von einem virtuellen Anwendungsserver als gehostete Anwendungen auf Benutzergeräten bereit. Wenn die Anwendungen wie unter [Microsoft Application Virtualization](#) beschrieben konfiguriert wurden, können sie Anwendungsbereitstellungsgruppen oder Desktop- und Anwendungsbereitstellungsgruppen hinzugefügt werden. Beachten Sie Folgendes:
 - Versionen vor App-V 5.0 werden nicht unterstützt.
 - Der App-V-Client 5,0 bietet keinen Offlinezugriff auf Anwendungen.
 - Verwenden Sie die nachfolgend aufgeführten Versionskombinationen für App-V- und XenDesktop-/XenApp-Komponenten.

App-V version	XenDesktop-/XenApp-Version	
	Delivery Controller	Virtual Delivery Agent (VDA)
5.0 R5TM	XenDesktop 7.0, 7.1, 7.5 XenApp 7.5	7.0, 7.1, 7.5
5.0 Service Pack 1	XenDesktop 7.0, 7.1, 7.5 XenApp 7.5	7.0, 7.1, 7.5
5.0 Service Pack 2	XenDesktop 7.0, 7.1, 7.5 XenApp 7.5	7.1, 7.5

Das folgende Diagramm zeigt die Anwendungsbereitstellung in Studio.



Nachdem die Anwendungen ausgewählt und installiert wurden, werden sie über gehostete Anwendungen und Desktopbereitstellungsgruppen bereitgestellt. Wenn ein Benutzer eine Anwendung anfordert, wird sie auf einer Maschine installiert und ausgeführt, die einer Anwendungsbereitstellungsgruppe oder Desktop- und Anwendungsbereitstellungsgruppe zugeordnet ist, der der Benutzer wie unter [Erstellen einer Bereitstellungsguppenanwendung](#) beschrieben zugewiesen wurde. Maschinen in einer Bereitstellungsgruppe (und die zugeordneten Anwendungen) werden den Benutzern entweder bei der Anmeldung nach dem Zufallsprinzip zugewiesen oder jedes Mal dem gleichen Benutzer bei seiner Anmeldung statisch zugewiesen. Die Maschinen, auf denen Anwendungen gehostet werden, sind wie folgt konfiguriert:

- Serverbetriebssystem: Gleichzeitiges Hosten von Desktops und Anwendungen von der gleichen Maschine für mehrere, gleichzeitig verbundene Benutzer.

- Desktopbetriebssystem: Bereitstellen eines Desktops oder einer Anwendungsgruppe von einer Maschine für jeweils einen Benutzer.
Da die Anwendungen auf Maschinen im Datencenter ausgeführt werden, wird die Verarbeitung auf Benutzergeräten auf ein Minimum reduziert. Durch das Hosten von Anwendungen auf Desktops im Datencenter wird die Anwendungsverwaltung vereinfacht und eine kostengünstige Lösung für die Anwendungsbereitstellung erzielt. Anwendungen und Desktops bleiben im Datencenter, so wird eine höhere Sicherheit und eine konsistente Benutzererfahrung gewährleistet.
- Remote-PC-Zugriff-Bereitstellungsgruppe: Benutzer melden sich remote an ihrem physischen PC an, auf dem die Anwendungen installiert sind. Citrix Receiver auf dem Clientgerät bietet Zugriff auf alle Anwendungen und Daten auf dem Remote-PC.

Hinweis: Um Anwendungen zu veröffentlichen oder Desktops mit Microsoft Remotedesktopdienst (RDS) zu hosten, müssen Sie Virtual Delivery Agent (VDA) für Serverbetriebssysteme auf dem Hostserver installieren.

Für die Benutzer ist die Interaktion mit den Anwendungen so, als ob sie lokal installiert wären. Um auf Anwendungen zuzugreifen, klicken Benutzer auf das Anwendungssymbol oder die Verknüpfung auf ihren Geräten und die Anwendung wird in High Definition angezeigt. Beim Abmelden werden die vom Benutzer vorgenommenen Änderungen verworfen, es sei denn, sie haben eine persönliche vDisk, einen statisch zugewiesenen Desktop oder beides.

Zusätzlich zur Bereitstellungsmethode wirken sich auch die Einstellungen der Anwendungseigenschaften auf die Benutzererfahrung aus. Diese Einstellungen werden konfiguriert, wenn Sie eine Anwendung einer neuen oder vorhandenen Bereitstellungsgruppe hinzufügen.

Erstellen einer Bereitstellungsgruppenanwendung

Sep 29, 2015

Mit dem Feature Anwendung erstellen in Studio wählen Sie Anwendungen aus, auf die die Benutzer beim Anmelden am Desktop zugreifen können. Studio erkennt die Anwendungen in den Masterimages bzw. Vorlagen. Sie können außerdem manuell benutzerdefinierte Anwendungen hinzufügen, die auf den Maschinen einer Bereitstellungsgruppe gespeichert sind. Die folgenden Anwendungstypen sind verfügbar:

- Auf Bereitstellungsgruppenmaschinen installierte Anwendungen: Der Assistent von Studio erkennt diese Anwendungen und stellt sie auf den Desktops der Bereitstellungsgruppe zur Verfügung. Die folgenden Anwendungstypen werden möglicherweise ermittelt und zur Verfügung gestellt:
 - 16-Bit- und 32-Bit-Anwendungen, die auf Computern mit Windows XP oder Windows 7 gehostet werden
 - Auf App-V-Servern gehostete Anwendungen
 - Auf Benutzermaschinen gehostete Anwendungen
- Benutzerdefinierte Anwendungen:
 - Alle Anwendungen, die nicht vom Assistenten ermittelt wurden, die jedoch eingeschlossen werden sollen.
 - Anwendungen, die auf dem Benutzergerät installiert sind. Sobald sie ausgewählt wurden, werden diese Anwendungen auf dem Desktop im Startmenü von Windows angezeigt.
- Offline-App-V-Anwendungen: Wenn Sie planen, für einen längeren Zeitraum ohne Verbindung zum Netzwerk zu arbeiten, dann können Sie über den Microsoft App-V-Client im Offlinemodus arbeiten. Folgen Sie den Anweisungen unter [How to Work Offline or Online with Application Virtualization](#).
Wenn der App-V 5-Client an dem Endpunkt installiert ist, von dem Sie Offline-App-V-Anwendungen aus anbieten möchten, werden die für den Benutzer über den App-V 5-Verwaltungsserver veröffentlichten Anwendungen dem Benutzer zugänglich gemacht.

Hinweis: Sie können Anwendungen für Remote-PC-Zugriff-Bereitstellungsgruppen erstellen.

1. Wählen Sie in Studio den Knoten Bereitstellungsgruppe und dann die Registerkarte Anwendungen aus und klicken Sie auf Anwendungen erstellen.
2. Wählen Sie auf der Seite Bereitstellungsgruppe eine Bereitstellungsgruppe zum Hosten der Anwendungen und klicken Sie dann auf Weiter.
3. Fügen Sie auf der Seite Anwendungen vorhandene Anwendungen dieser Bereitstellungsgruppe aus der Liste der Anwendungen hinzu, die von Studio ermittelt und angezeigt werden.
Hinweis: Der zugewiesene Benutzer bzw. die zugewiesene Gruppe muss Mitglied der Bereitstellungsgruppe sein, der die Anwendung zugewiesen wurde, um die Anwendung anzuzeigen oder darauf zuzugreifen.
4. Wenn Sie eine benutzerdefinierte Anwendung hinzufügen möchten, klicken Sie auf Anwendungen manuell hinzufügen und klicken Sie dann auf Weiter. Geben Sie auf der Seite Anwendung manuell hinzufügen die folgenden Informationen über die Anwendung an:
 - Pfad zu der ausführbaren Datei
 - Optionale Befehlszeilenparameter
 - Verzeichnis
 - Anzeigenamen für Administrator und Benutzer: Name der Benutzer und Name, der nur Administratoren angezeigt wird
 - Speicherort: Pfad zur ausführbaren Datei der Anwendung, Arbeitsverzeichnis und optional ein Befehlszeilenargument.
 - Sichtbarkeit beschränken: Gibt an, für welche Benutzer oder Benutzergruppen die Anwendung angezeigt wird.
Hinweis: Der zugewiesene Benutzer bzw. die zugewiesene Gruppe müssen Mitglied der Bereitstellungsgruppe sein, der

die Anwendung zugewiesen wurde. Sonst können Benutzer die Anwendung nicht anzeigen oder auf sie zugreifen.

- Dateitypzuordnung: Legt fest, welche Typen von Dateien automatisch in der Anwendung geöffnet werden.

5. Überprüfen Sie auf der Seite Zusammenfassung alle Informationen.

6. Wiederholen Sie die Schritte zum Erstellen von Anwendungen für zusätzliche Anwendungsbereitstellungsgruppen oder für Anwendungs- und Desktopbereitstellungsgruppen.

Microsoft Application Virtualization

Sep 29, 2015

Mit Microsoft Application Virtualization (App-V) können Sie Anwendungen als Dienste bereitstellen, aktualisieren und unterstützen. Benutzer können auf Anwendungen zugreifen, ohne sie auf ihren Geräten installieren zu müssen. Anwendungen und Benutzereinstellungen werden beibehalten, unabhängig davon, ob Benutzer online oder offline sind.

Dieses Release unterstützt App-V 5.0.

Der Offlinezugriff auf Anwendungen wird von App-V-Client 5.0 nicht unterstützt.

Der App-V-Client 4.6.2 wird nicht mehr unterstützt.

App-V und Microsoft User State Virtualization (USV) ermöglichen den Zugriff auf Anwendungen und Daten unabhängig vom Standort oder von der Internetverbindung.

Die Unterstützung der App-V-Integration umfasst die Verwendung von SMB-Freigaben für Anwendungen. Das HTTP-Protokoll wird nicht unterstützt.

Microsoft App-V-Komponenten

Informationen zu Systemanforderungen finden Sie unter [App-V 5.0 Supported Configurations](#).

Mit App-V werden einzelne Anwendungen von lokal installierten Produkten in zentral verwaltete Dienste transformiert. Anwendungen sind nahtlos verfügbar, ohne dass Vorkonfigurationen oder Änderungen an den Einstellungen des Betriebssystems vorgenommen werden müssen. App-V enthält die folgenden Komponenten:

- **Verwaltungsserver:** Bietet eine zentrale Konsole zum Verwalten der App-V 5.0-Infrastruktur und zum Bereitstellen von virtuellen Anwendungen sowohl für App-V-Desktop Client als auch für einen Remotedesktopdienste-Client. Der App-V-Verwaltungsserver führt das vom Administrator benötigte Authentifizieren, Anfordern und Bereitstellen von Sicherheit, Messungen, Überwachung und Sammeln von Daten durch. Der Server verwendet Active Directory und unterstützende Tools zum Verwalten von Benutzern und Anwendungen.
- **Veröffentlichungsserver:** Bietet App-V-Clients Anwendungen für bestimmte Benutzer und hostet das virtuelle Anwendungspaket für das Streaming. Die Pakete werden vom Verwaltungsserver abgerufen.
- **Client:** Ruft virtuelle Anwendungen ab, veröffentlicht die Anwendungen auf dem Client und erstellt und verwaltet automatisch virtuelle Umgebungen zur Laufzeit auf Windows-Geräten. Der App-V-Client wird auf dem Virtual Desktop Agent installiert und speichert benutzerspezifische, virtuelle Anwendungseinstellungen, wie Registrierungs- und Dateiänderungen, in den Benutzerprofilen.

Studio-Komponenten

Mit den folgenden Komponenten können Sie Benutzern App-V-Anwendungen in Studio zur Verfügung stellen:

- **Studio App-V-Veröffentlichung:** Konfiguriert den Zugriff auf App-V-Veröffentlichungs- und Verwaltungsserver, sodass Studio App-V-Anwendungen erkennen und verwenden kann.
- **Maschinenkataloge:** Sammlungen physischer Computer und virtueller Maschinen, auf denen Anwendungen installiert sind. Die Maschinen und ihre Ressourcen werden Benutzern über Bereitstellungsgruppen zugewiesen. Während der Erstellung des Maschinenkatalogs muss der App-V-Client auf dem Masterimage installiert werden und über Einstellungen wie `ShareContentStoreMode` und `EnablePackageScripts` konfiguriert werden.
Hinweis: Sie brauchen den App-V-Veröffentlichungsserver nicht im Masterimage zu konfigurieren, da die Konfiguration beim Anwendungsstart erfolgt.
- **Bereitstellungsgruppen:** Bieten bestimmten Benutzergruppen, wie Abteilungen oder Teams, Zugriff auf Anwendungen.
- **Anwendungen:** Werden Bereitstellungsgruppen zugewiesen und auf diese Weise Benutzern zugänglich gemacht.

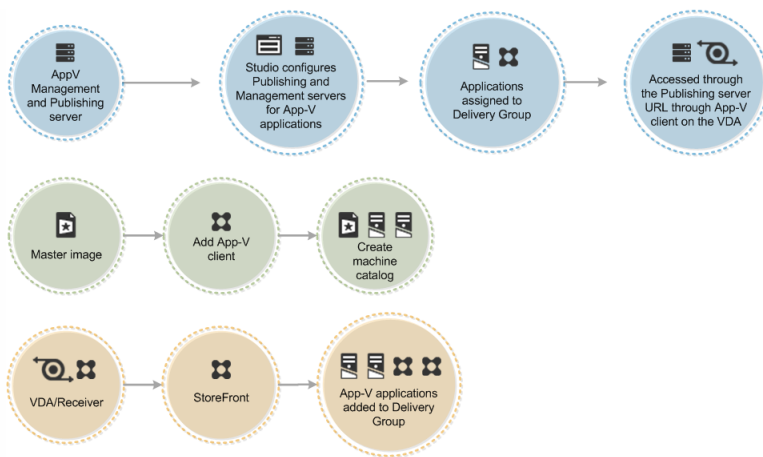
Mit App-V stellen Sie Anwendungen von einem virtuellen Anwendungsserver auf einem Benutzergerät bereit.

Dieses Release unterstützt App-V 5.

Für das Bereitstellen dieser Typen von Anwendungen ist Folgendes erforderlich:

1. Bereitstellen von App-V, wie in den Anweisungen in der Microsoft TechNet Library unter <http://technet.microsoft.com/en-us/virtualization/hh710199> beschrieben
2. Veröffentlichen der App-V-Anwendungen auf dem App-V-Verwaltungsserver Konfigurieren von Einstellungen, u. a. Berechtigungen und Dateitypzuordnung Diese Einstellungen sind bereits vorhanden, wenn Sie App-V bereits bereitgestellt haben.
3. Installieren Sie den App-V-Client auf VDAs.
4. Installieren Sie den App-V-Client auf dem Masterimage für Studio-Maschinenkataloge.
Hinweis: Während der Erstellung des Maschinenkatalogs muss der App-V-Client auf dem Masterimage installiert werden und über Einstellungen wie `ShareContentStoreMode` und `EnablePackageScripts` konfiguriert werden. Sie brauchen den App-V-Veröffentlichungsserver nicht im Masterimage zu konfigurieren, da die Konfiguration beim Anwendungsstart erfolgt.
5. Fügen Sie während der Sitekonfiguration in Studio die Adressen des App-V-Veröffentlichungsservers und des -Verwaltungsservers hinzu. Diese Server werden automatisch von den Bereitstellungsgruppen in der Site abgerufen.
6. Stellen Sie mit Studio eine Verbindung mit den App-V-Servern her, wie unter [Microsoft Application Virtualization](#) beschrieben.
7. Stellen Sie mit Studio App-V-Anwendungen über Anwendungsbereitstellungsgruppen bereit, wie in [Erstellen einer Bereitstellungsgruppenanwendung](#) beschrieben.
8. Die Anwendungen können jetzt über Citrix Receiver gestartet werden; der Zugriff erfolgt über StoreFront.

In der folgenden Abbildung werden die Prozesse für das Bereitstellen von App-V-Anwendungen dargestellt.



Zum Ändern der Veröffentlichungsservereinstellungen auf einem VDA empfiehlt Citrix die Verwendung der SDK-Cmdlets auf dem Controller.

Zum Anzeigen der Veröffentlichungsservereinstellungen geben Sie Folgendes ein:

Get-CtxAppvServerSetting -AppVPublishingServer

Das folgende Cmdlet ändert die Veröffentlichungsservereinstellungen auf dem Delivery Controller. Nicht alle Parameter sind obligatorisch.

Set-CtxAppvServerSetting -AppVPublishingServer -UserRefreshOnLogon -UserRefreshEnabled -UserRefreshInterval -UserRefreshIntervalUnit -GlobalRefreshOnLogon -GlobalRefreshEnabled -GlobalRefreshInterval -GlobalRefreshIntervalUnit
Damit sichergestellt wird, dass App-V 5-Anwendungen richtig gestartet werden, geben Sie in PowerShell das folgende SDK-Cmdlet ein:

Set-CtxAppvServerSetting -UserRefreshOnLogon 0

Hinweis: Wenn Sie zuvor GPO-Richtlinieneinstellungen für die Verwaltung der Veröffentlichungsservereinstellungen verwendet haben, werden die App-V-Integrationseinstellungen, einschließlich der vorherigen Cmdlet-Einstellungen, von den GPO-Einstellungen überschrieben. Dies kann dazu führen, dass der Start der App-V-Anwendung fehlschlägt. Citrix empfiehlt, dass Sie alle GPO-Richtlinieneinstellungen entfernen und die gleichen Einstellungen mit dem SDK konfigurieren, wie in der Dokumentation zum SDK-Cmdlet erläutert.

Sie können App-V-Anwendungen (von Serverbetriebssystem- und Desktopbetriebssystem-Bereitstellungsgruppen starten, wie in den folgenden Anwendungsfällen beschrieben:

- Über Citrix Receiver
- Vom Startmenü
- Über den App-V-Client und Citrix Receiver
- Gleichzeitig von mehreren Benutzern auf mehreren Geräten
- Über Citrix StoreFront

Beachten Sie Folgendes:

- Geänderte App-V-Anwendungseigenschaften werden implementiert, wenn die Anwendung gestartet wird. Beispiel: Bei Anwendungen mit einem geänderten Anzeigenamen oder einem angepassten Symbol wird die Modifikation angezeigt, wenn Benutzer die Anwendung starten.
- Es besteht keine Änderung in der Leistung von App-V-Anwendungen, wenn ein Desktop und eine Anwendungsbereitstellungsgruppe in eine Anwendungsbereitstellungsgruppe konvertiert wird.
- Es wird nur eine serverbasierte App-V-Bereitstellung unterstützt, in der ein Administrator mit einem App-V-Verwaltungsserver und -Veröffentlichungsserver App-V-Anwendungen verwaltet.

1. Wählen Sie in Studio im Knoten "Konfiguration" App-V-Veröffentlichung aus.
2. Wählen Sie App-V-Veröffentlichung hinzufügen.
3. Geben Sie im Dialogfeld App-V-Einstellungen die URLs für die folgenden Server ein:
 - App-V-Verwaltungsserver
 - App-V-Veröffentlichungsserver
4. Wählen Sie Verbindung testen, um sicherzustellen, dass Sie auf den Server zugreifen können, und klicken Sie dann auf Speichern.

Sie können App-V-Server hinzufügen oder ändern, die Anwendungsanzeige aktualisieren oder Anwendungen löschen.

1. Wählen Sie im Knoten "Konfiguration" App-V-Veröffentlichung aus.
2. Sie können die folgenden Aktionen ausführen:
 - App-V-Veröffentlichungseinstellungen bearbeiten: Hinzufügen und Bearbeiten von App-V-Verwaltungs- und App-V-Veröffentlichungsservern.
 - App-V-Veröffentlichungseinstellungen entfernen: Löschen von App-V-Verwaltungs- und App-V-Veröffentlichungsservern.
 - App-V-Anwendungen aktualisieren: Aktualisieren der App-V-Anwendungsanzeige. Wenn Anwendungen nicht mehr verfügbar sind oder keine Verbindung zum Server hergestellt werden kann, wird dies entsprechend angezeigt.

Da die App-V-Anwendungen nun verfügbar sind, weisen Sie sie den Benutzern wie unter [Erstellen einer Bereitstellungsgruppenanwendung](#) beschrieben zu.

Problembehandlung bei Studio

Verbindungstest schlägt fehl

Wenn Sie die Adresse des App-V-Verwaltungsservers und des Veröffentlichungsservers in Studio festlegen und beim Testen der Verbindung ein Fehler auftritt, prüfen Sie Folgendes:

1. App-V-Server ist eingeschaltet: Senden Sie entweder einen Ping-Befehl oder prüfen Sie in IIS-Manager, ob für jeden Server als Zustand "Gestartet" und "Wird ausgeführt" angezeigt wird.

- PowerShell Remoting ist auf dem App-V-Server aktiviert. Ist dies nicht der Fall, folgen Sie den Anweisungen unter <http://technet.microsoft.com/en-us/magazine/ff700227.aspx>.
- Der App-V-Server wurde Active Directory hinzugefügt.
Sind Studio-Maschine und App-V-Server in verschiedenen Active Directory-Domänen, zwischen denen keine Vertrauensbeziehung besteht, führen Sie über die PowerShell-Konsole auf der Studio-Maschine `winrm/config/client @{TrustedHosts="<App-V server FQDN>"}` aus. Wird "TrustedHosts" über das Gruppenrichtlinienobjekt verwaltet, wird eine Fehlermeldung angezeigt, die besagt, dass die Konfigurationseinstellung "TrustedHosts" nicht geändert werden kann, da sie von Richtlinien gesteuert wird, und dass die Richtlinie auf "Nicht konfiguriert" festgelegt werden müsste, damit die Konfigurationseinstellung geändert werden kann. Wenn diese Meldung angezeigt wird, fügen Sie einen Eintrag für den App-V-Servernamen in der TrustedHosts-Richtlinie im Gruppenrichtlinienobjekt hinzu (Administrative Vorlagen > Windows-Komponenten > Windows-Remoteverwaltung (WinRM) > WinRM-Client).
- Der Studio-Administrator ist gleichzeitig App-V-Serveradministrator.
- Auf dem App-V-Server ist die Dateifreigabe aktiviert. Geben Sie hierfür in Windows-Explorer oder über den Befehl "Ausführen" Folgendes ein: `\\<App-V-Server FQDN>`
- Der App-V-Server hat dieselben Dateifreigabeberechtigungen wie der App-V-Administrator. Fügen Sie hierzu auf dem App-V-Server einen Eintrag für `\\<App-V-Server FQDN>` in "Gespeicherte Benutzernamen und Kennwörter" hinzu und geben Sie die Anmeldeinformationen des Benutzers ein, der Administratorberechtigungen auf dem App-V-Server hat. Weitere Informationen finden Sie unter <http://support.microsoft.com/kb/306541>.

App-V-Anwendungserkennung schlägt fehl

Überprüfen Sie Folgendes, wenn die App-V-Anwendungserkennung fehlschlägt:

- Der Studio-Administrator ist ein Administrator auf dem App-V-Verwaltungsserver.
- Der App-V-Verwaltungsserver wird ausgeführt. Prüfen Sie dies, indem Sie den IIS-Manager öffnen und sicherstellen, dass sich der Server im Zustand "Gestartet" und "Ausgeführt" befindet.
- PowerShell-Remoting ist auf dem App-V-Verwaltungsserver und auf dem Veröffentlichungsserver aktiviert. Wenn einer der beiden Server nicht aktiviert ist, folgen Sie den Anweisungen unter <http://technet.microsoft.com/en-us/magazine/ff700227.aspx>, um den Server zu aktivieren.
- Pakete haben entsprechende Sicherheitsberechtigungen, sodass der Studio-Administrator Zugriff hat.

Problembehandlung, wenn der App-V-Start fehlschlägt

Prüfen Sie Folgendes, wenn App-V-Anwendungen nicht gestartet werden:

- Der Veröffentlichungsserver wird ausgeführt. Prüfen Sie dies, indem Sie den IIS-Manager öffnen und sicherstellen, dass sich der Server im Zustand "Gestartet" und "Ausgeführt" befindet.
 - App-V-Pakete haben entsprechende Sicherheitsberechtigungen, sodass Benutzer Zugriff haben.
 - Prüfen Sie Folgendes auf dem VDA:
 - Stellen Sie sicher, dass Temp auf den richtigen Speicherort verweist und dass genügend Speicherplatz im Verzeichnis Temp verfügbar ist.
 - Stellen Sie sicher, dass der App-V-Client installiert ist, und zwar mindestens in Version 5.0.
 - Stellen Sie sicher, dass Sie Administratorrechte haben, führen Sie `Get-AppvClientConfiguration` aus und vergewissern Sie sich, dass "EnablePackageScripts" auf 1 festgelegt ist. Wenn es nicht auf "1" gesetzt ist, führen Sie `Set-AppvClientConfiguration -EnablePackageScripts $true` aus.
Citrix empfiehlt, dass Sie diesen Schritt beim Erstellen eines Masterimages ausführen, sodass alle VDAs, die vom Masterimage erstellt werden, die richtige Konfiguration haben.
 - Öffnen Sie den Registrierungs-Editor (regedit) und navigieren Sie zu `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Citrix\Appv`. Stellen Sie sicher, dass der Schlüssel "AppVServers" den folgenden Wert hat: `AppVManagementServer+metadata:PublishingServer`. Beispiel:
`http://xmas-demo-appv.blrstrm.com+0+0+0+1+1+1+0+1`; `http://xmas-demo-appv.blrstrm.com:8082`
 - Stellen Sie sicher, dass CtxAppVCOMAdmin über Administratorrechte verfügt. Bei der Installation von VDA wird normalerweise CtxAppVCOMAdmin erstellt und der lokalen Administratorgruppe auf der VDA-Maschine hinzugefügt. Abhängig von der Active Directory-Richtlinie verliert der Benutzer u. U. die administrative Zuordnung.
Führen Sie `compmgmt.msc` aus und navigieren Sie zu Lokale Benutzer und Gruppen > Benutzer. Wenn CtxAppVCOMAdmin kein Administrator ist, passen Sie die Gruppenrichtlinie an oder wenden Sie sich an den Administrator, damit das Benutzerkonto seine administrative Zuordnung behält.
 - Auf dem Masterimage, auf dem der App-V-Client installiert ist, muss der Administrator die "PowerShell ExecutionPolicy" auf RemoteSigned festlegen. Diese Einstellung ist erforderlich, da das von Microsoft zur Verfügung gestellte App-V-Clientmodul nicht signiert ist. Mit der Einstellung kann PowerShell unsigned lokale Skripts und Cmdlets ausführen. Legen Sie die ExecutionPolicy mit einer dieser Methoden fest:
 - Melden Sie sich als Administrator an und führen Sie den PowerShell-Befehl `Set-ExecutionPolicy RemoteSigned` aus.
 - Navigieren Sie von den Gruppenrichtlinieneinstellungen zu "Computerkonfiguration > Richtlinien > Administrative Vorlagen > Windows-Komponenten > Windows PowerShell > Skriptausführung aktivieren".
 - Überprüfen Sie die Veröffentlichungsserver:
 - Führen Sie `Get-AppvPublishingServer *` aus, damit die Liste der Veröffentlichungsserver angezeigt wird.
 - Überprüfen Sie, ob `UserRefreshOnLogon` auf "False" festgelegt ist. Wenn nicht "False" festgelegt ist, schlägt der erste App-V-Anwendungsstart normalerweise fehl.
 - Führen Sie mit Administratorprivilegien `Set-AppvPublishingServer` aus und stellen Sie "UserRefreshOnLogon" auf "False" ein.
- Hinweis: Ab XenDesktop 7.1 bzw. XenApp 7.5 können Sie mit SDK-Cmdlets den Parameter "UserRefreshOnLogon" (und andere Parameter für Veröffentlichungsserver) anzeigen und festlegen.

Wenn Sie mit diesen Schritten die Probleme nicht beheben können, sollte der Administrator die Protokolle aktivieren und überprüfen.

Nach dem Upgrade eines App-V-Pakets

Wenn Sie ein Upgrade eines App-V-Pakets mit über Studio veröffentlichten Anwendungen durchführen, löschen Sie die alten Anwendungen. Dadurch werden diese aus der Anwendungsbereitstellungsgruppe gelöscht. Anschließend führen Sie eine Anwendungserkennung durch und fügen die neuen Anwendungen der Bereitstellungsgruppe hinzu. Beispiel: Wenn Sie Word 2010 und Excel 2010 aus einem Office 2010-App-V-Paket veröffentlicht haben und das Paket dann auf Office 2013 aktualisieren, entfernen Sie Word 2010 und Excel 2010, führen die Erkennung durch und fügen dann Word 2013 und Excel 2013 hinzu.

Aktivieren von Protokollen

Aktivieren Sie Protokolle in Studio und im VDA zur Problembehandlung von App-V.

- Erstellen Sie den Ordner `C:\CtxAppVLogs`.
- Gehen Sie zu `"C:\Programme\Citrix\StudioAppVIntegration\SnapiN\Citrix.Appv.Admin.V1"` und öffnen Sie als Administrator `CtxAppvCommon.dll.config` in einem Text-Editor, z. B. Editor. Heben Sie die Auskommentierung der folgenden Zeile auf:

Die Protokolle stehen an diesem Speicherort zur Verfügung.

1. Erstellen Sie den Ordner C:\CtxAppvLogs.
2. Gehen Sie zu C:\Programme\Citrix\Virtual Desktop Agent und öffnen Sie als Administrator CtxAppvCommon.dll.config in einem Text-Editor, z. B. Editor.
3. Heben Sie die Auskommentierung der folgenden Zeile in CtxAppvCommon.dll.config auf:
4. Heben Sie die Auskommentierung der folgenden Zeile auf und stellen Sie den Wert auf 1 ein, wie im folgenden Beispiel:
 - Alle Protokolle in Verbindung mit der Konfiguration sind unter C:\CtxAppvLogs verfügbar.
 - Die Anwendungsstartprotokolle sind wie folgt verfügbar:
 - XenDesktop 7.1, XenDesktop 7.5 oder XenApp 7.5: %LOCALAPPDATA%\Citrix\CtxAppvLogs.
 - XenDesktop 7.0: %LocalAppData%\temp\CtxAppVLogs\
 - LOCALAPPDATA wird in den lokalen Ordner des angemeldeten Benutzers aufgelöst. Prüfen Sie den lokalen Ordner des Benutzers, für den der Anwendungsstart fehlschlägt:
 - Ab XenDesktop 7.1 bzw. XenApp 7.5: %LocalAppData%\Citrix\CtxAppVLogs\.
 - XenDesktop 7.0: %LocalAppData%\temp\CtxAppVLogs\
5. Starten Sie als Administrator den Brokerdienst oder die VDA-Maschine neu, damit die Protokollierung gestartet wird.

Lokaler App-Zugriff

Sep 29, 2015

Durch lokalen App-Zugriff werden die lokal installierten Windows-Anwendungen von Benutzern problemlos in eine gehostete Desktopumgebung integriert, ohne dass ein Wechsel zwischen Computern nötig ist. Lokaler App-Zugriff ermöglicht Folgendes:

- Direkter Zugriff von virtuellen Desktops auf Anwendungen, die lokal auf einem Laptop, PC oder einem anderen Gerät installiert sind

Bereitstellung einer flexiblen Anwendungsbereitstellungslösung Wenn Benutzer lokale Anwendungen haben, die Sie nicht virtualisieren können oder die IT nicht verwaltet, verhalten sich diese Anwendungen weiterhin so, als ob sie auf einem virtuellen Desktop installiert wären.

- Eliminieren Sie Doppelhoplatenz bei separat vom virtuellen Desktop gehosteten Anwendungen, indem Sie die Verknüpfung mit der veröffentlichten Anwendung auf das Windows-Gerät des Benutzers platzieren.
- Unter anderem können die folgenden Anwendungen verwendet werden:
 - Videokonferenzsoftware, z. B. GoToMeeting.
 - Spezial- oder Nischenanwendungen, die noch nicht virtualisiert sind.
 - Anwendungen und Peripheriegeräte, die andernfalls riesige Datenmengen von einem Benutzergerät zum Server und zurück zum Benutzergerät senden würden, wie DVD-Brenner und TV-Tuner.

Führen Sie die folgenden Schritte zum Einrichten von lokaler App-Zugriff aus.

- Aktivieren Sie das Feature für den lokalen App-Zugriff wie unter [Bereitstellen von Zugriff auf lokale Anwendungen](#) beschrieben.
- Aktivieren Sie den lokalen App-Zugriff auf Clientmaschinen mit Receiver wie unter [Aktivieren von lokalem App-Zugriff auf Clientmaschinen mit Receiver](#) beschrieben.

Anforderungen für lokalen App-Zugriff

Der lokale App-Zugriff wird für Desktopbetriebssystemmaschinen, Serverbetriebssystemmaschinen und Remote-PC-Zugriff-Desktops unterstützt.

Zum Aktivieren des lokalen App-Zugriffs muss die Hostingumgebung folgende Komponenten enthalten:

- XenDesktop 7.x und XenApp 7.5
- StoreFront
- Betriebssysteme für gehostete Desktops:
 - Windows Server 2012 R2
 - Windows Server 2012
 - Windows Server 2008 R2
 - Windows 7 (32 Bit und 64 Bit)
 - Windows 8 (32 Bit und 64 Bit)
 - Windows 8.1 (32 Bit und 64 Bit)
- Betriebssysteme für Clients:
 - Windows XP SP3 (32 Bit)
 - Windows 7 (32 Bit und 64 Bit)
 - Windows 8 (32 Bit und 64 Bit)

- Windows 8.1 (32 Bit und 64 Bit)
- Webbrowser (nur die hier aufgeführten werden unterstützt):
 - Internet Explorer 8, 9 und 10
 - Mozilla Firefox 3.5 bis 21.0
 - Google Chrome 10
- Citrix Receiver 4.0

Einschränkungen

- Lokaler App-Zugriff ist nur für virtuelle Desktops im Vollbildmodus unter Einbeziehung aller Monitore gedacht:
 - Die Benutzererfahrung könnte beeinträchtigt werden, wenn lokaler App-Zugriff auf einem virtuellen Desktop verwendet wird, der im Fenstermodus ausgeführt wird oder der nicht auf allen Monitoren ausgeführt wird.
 - Bei der Verwendung mehrerer Monitore ist der maximierte Monitor der Standarddesktop für alle Anwendungen, die in der Sitzung gestartet werden, selbst wenn die nachfolgenden Anwendungen normalerweise auf dem anderen Monitor starten würden.
 - Das Feature ist für die Verwendung mit einem VDA konzipiert; es ist keine Integration mit mehreren gleichzeitigen VDAs möglich.
- Einige Anwendungen zeigen unerwartetes Verhalten, das folgende Auswirkungen auf Benutzer haben könnte:
 - Benutzer können die Laufwerksbuchstaben verwechseln, z. B. das lokale C:-Laufwerk mit dem virtuellen C:-Desktoplaufwerk.
 - Auf virtuellen Desktops verfügbare Drucker sind nicht für die lokalen Anwendungen verfügbar.
 - Anwendungen, die erweiterte Berechtigungen erfordern, können nicht als clientgehostete Anwendungen gestartet werden.
 - Keine spezielle Behandlung von Anwendungen mit einer Instanz (z. B. Windows Media Player).
 - Lokale Anwendungen werden mit dem Windows-Design des lokalen Computers angezeigt.
 - Vollbildanwendungen werden nicht unterstützt. Dies schließt Anwendungen ein, die im Vollbildmodus geöffnet werden, z. B. PowerPoint-Bildschirmpräsentationen oder Fotoanzeigen, die den gesamten Desktop ausfüllen.
 - Lokaler App-Zugriff kopiert die Eigenschaften der lokalen Anwendung wie die vorhandenen Verknüpfungen auf dem Clientdesktop und im Startmenü des VDA. Allerdings werden keine anderen Eigenschaften, wie Tastenkombinationen und schreibgeschützte Attribute, kopiert.
 - Anwendungen, die die Reihenfolge der überlappenden Fenster anpassen, können unvorhersehbare Ergebnisse verursachen. Beispielsweise könnten einige Fenster ausgeblendet werden.
 - Verknüpfungen, einschließlich Arbeitsplatz, Papierkorb, Systemsteuerung, Netzlaufwerkverknüpfungen und Ordnerverknüpfungen werden nicht unterstützt.
 - Die folgenden Dateitypen und Dateien werden nicht unterstützt: benutzerdefinierte Dateitypen, Dateien ohne zugeordnete Programme, ZIP-Dateien und ausgeblendete Dateien.
 - Taskleistengruppierung wird nicht für gemischte 32-Bit und 64-Bit clientgehostete Anwendungen und VDA-Anwendungen unterstützt, z. B. die Gruppierung von 32-Bit-Versionen lokaler Anwendungen mit 64-Bit-VDA-Anwendungen und umgekehrt.
 - Anwendungen können nicht über COM gestartet werden. Beispiel: Wenn Sie auf ein eingebettetes Office-Dokument in einer Office-Anwendung klicken, wird der Prozessstart nicht erkannt und die Integration der lokalen Anwendung schlägt fehl.
- Die URL-Umleitung unterstützt nur explizite URLs, d. h. solche, die in der Adressleiste des Browsers angezeigt werden oder mit der browserinternen Suchfunktion gefunden wurden (je nach Browser).
- Die URL-Umleitung funktioniert nur mit Desktopsitzungen und nicht mit Anwendungssitzungen.
- Benutzer haben keine Berechtigung, im lokalen Desktopordner in einer VDA-Sitzung neue Dateien zu erstellen.

Bereitstellen von Zugriff auf lokale Anwendungen

Für den Zugriff auf lokale Anwendungen müssen Sie das Feature für den lokalen App-Zugriff aktivieren, die nicht standardmäßig für die Host- und Clientumgebung aktiviert ist.

Bereitstellen von Zugriff auf alle lokalen Anwendungen

Aktivieren Sie dieses Feature, indem Sie den Anweisungen unter

— *Aktivieren von lokalem App-Zugriff mit Richtlinieneinstellungen*

und

— *Aktivieren von lokalem App-Zugriff in Receiver*

folgen.

Nachdem Sie diese Schritte ausgeführt haben, ist das Feature für den lokalen App-Zugriff aktiviert und die Verknüpfungen des Benutzers werden auf dem Desktop des VDA angezeigt.

Bereitstellen von Zugriff auf veröffentlichte Anwendungen

Verwenden Sie dieses Verfahren, wenn Sie Zugriff nur auf veröffentlichte Anwendungen bereitstellen möchten.

1. Öffnen Sie den Registrierungs-Editor auf der Maschine, auf der Studio ausgeführt wird, und navigieren Sie zu folgendem Pfad: HKLM\Software\Wow6432Node\Citrix\DesktopStudio.
2. Fügen Sie den folgenden Registrierungseintrag vom Typ REG_DWORD hinzu: ClientHostedAppsEnabled und fügen Sie den Wert 1 hinzu, um den lokalen App-Zugriff zu aktivieren. (Durch die Verwendung von 0 wird der lokale App-Zugriff deaktiviert).
3. Starten Sie die Maschine neu und starten Sie dann Studio neu, damit die Änderungen angewendet werden.
4. Veröffentlichen Sie die Anwendungen vom lokalen App-Zugriff wie unter
— *Veröffentlichen lokaler Anwendungen in Studio*
erläutert.
5. Aktivieren Sie lokalen App-Zugriff in Receiver wie unter
— *Aktivieren von lokalem App-Zugriff in Receiver*
erläutert.
6. Legen Sie für die Richtlinie "Lokalen App-Zugriff zulassen" die Einstellung Aktiviert fest. Anweisungen hierzu finden Sie unter
— *Aktivieren von lokalem App-Zugriff mit Richtlinieneinstellungen*

Veröffentlichen lokaler Anwendungen in Studio

1. Wählen Sie in Studio im linken Bereich den Knoten "Bereitstellungsgruppen" aus und wählen Sie dann im Hauptbereich die Registerkarte "Anwendungen" aus. Klicken Sie im Aktionsbereich auf Clientgehostete Anwendung erstellen. Der Assistent zum Erstellen einer clientgehosteten Anwendung wird angezeigt.
2. Wählen Sie auf der Seite Bereitstellungsgruppen die Bereitstellungsgruppen für den Zugriff auf die lokale Anwendung aus.
Hinweis: Der lokale App-Zugriff ist nur für Desktop-Bereitstellungsgruppen verfügbar. Er ist nicht für Anwendungsbereitstellungsgruppen verfügbar.
3. Geben Sie auf der Seite "Speicherort" den vollständigen Pfad der ausführbaren Datei der Anwendung ein, wie auf der lokalen Maschine im Feld "Ausführbare Programmdatei".
Hinweis: Stellen Sie sicher, dass der Pfad richtig ist. Ansonsten ist die Anwendung nicht für Benutzer sichtbar.

4. Fügen Sie auf der Seite Benutzer die Benutzer hinzu, denen Sie Zugriff auf die lokale, zu veröffentlichende Anwendung geben möchten.
Hinweis: Sie können entweder einzelne Benutzer oder eine Gruppe aus Active Directory auswählen.
5. Legen Sie auf der Seite Verknüpfung fest, ob die Verknüpfung mit der lokalen Anwendung auf dem virtuellen Desktop im Startmenü, auf dem Desktop oder in beiden angezeigt wird.
6. Geben Sie auf der Seite Inhaltsumleitung die Dateitypzuordnung (FTA) der Anwendung ein. Standardmäßig werden alle Dateitypen der Anwendung zugeordnet, wenn Sie Dateitypen zur Laufzeit vom Computer des Benutzers abrufen auswählen. Ansonsten können Sie die Dateitypen explizit angeben, indem Sie Datentypen explizit festlegen auswählen und dann bestimmte Erweiterungen hinzufügen.
Zum Beispiel ordnet sich Microsoft Word standardmäßig alle mit Word verknüpften Erweiterungen (.doc, .docx, .rtf usw.) zu. Bei Wahl von "Dateitypen zur Laufzeit vom Computer des Benutzers abrufen" werden alle Erweiterungen ausgewählt. Wenn Sie jedoch die Dateitypen, die der Anwendung zugeordnet werden, einschränken möchten (z. B. nur .doc), wählen Sie "Datentypen explizit festlegen".
7. Übernehmen Sie auf der Seite Name die Standardwerte.
8. Überprüfen Sie auf der Seite Zusammenfassung die von Ihnen angegebenen Informationen und klicken Sie auf Fertig stellen.

Aktivieren von lokalem App-Zugriff in Receiver

Folgen Sie den Anweisungen unter [Aktivieren von lokalem App-Zugriff auf Clientmaschinen mit Receiver](#), um die Konfiguration für den lokalen App-Zugriff abzuschließen.

Aktivieren von lokalem App-Zugriff mit Richtlinienereinstellungen

Verwenden Sie die Richtlinie "Lokaler App-Zugriff" für Bereitstellungsgruppen, indem Sie die Richtlinie auf **Aktiviert** einstellen. Mit dieser Einstellung überlässt der VDA dem Client die Entscheidung darüber, ob Verknüpfungen für vom Administrator veröffentlichte Anwendungen und Verknüpfungen für den lokalen App-Zugriff in der Sitzung aktiviert sind.

Bei Einstellung auf **Deaktiviert** sind sowohl vom Administrator veröffentlichte Anwendungen als auch Verknüpfungen für den lokalen App-Zugriff für den VDA deaktiviert.

Diese Richtlinie gilt für den gesamten Computer als auch für die URL-Umleitungsrichtlinie. Weitere Informationen finden Sie unter [Einstellungen der Richtlinie "Lokaler App-Zugriff"](#).

Konfigurieren der URL-Umleitung

Anstelle der Windows-Registrierung können Sie zum Einrichten der URL-Umleitung Richtlinienereinstellungen verwenden, indem Sie die Richtlinien für die URL-Umleitungspositivliste und die URL-Umleitungssperreliste in Studio konfigurieren. Ausführliche Anweisungen zum Konfigurieren der URL-Umleitung finden Sie unter [Verwenden von URL-Umleitung zum Starten von lokalen Anwendungen](#).

Weitere Informationen über diese Richtlinie finden Sie unter [Einstellungen der Richtlinie "Lokaler App-Zugriff"](#).

Konfigurieren von Abmelden und Trennen für den lokalen App-Zugriff

Sie können das Verhalten lokaler Anwendungen konfigurieren, wenn sich der Benutzer abmeldet oder die Verbindung zum virtuellen Desktop trennt.

1. Öffnen Sie auf dem gehosteten Desktop den Registrierungs-Editor und navigieren Sie zu folgendem Pfad:

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\Client Hosted Apps\Policies\Session State.

2. Fügen Sie den Registrierungseintrag Terminate vom Typ REG_DWORD hinzu und stellen Sie den Wert folgendermaßen ein:

- 1 : Lokale Anwendungen werden weiterhin ausgeführt, wenn sich ein Benutzer abmeldet oder die Verbindung zum virtuellen Desktop trennt.
- 3 : Lokale Anwendungen werden geschlossen, wenn sich ein Benutzer abmeldet oder die Verbindung zum virtuellen Desktop trennt.

Hinweis: Wenn der Wert auf 1 eingestellt ist, werden die lokalen Anwendungen bei erneuter Verbindung mit dem virtuellen Desktop aus der getrennten Sitzung wieder integriert, wenn sie noch in der lokalen Umgebung verfügbar sind

Funktionsinteraktion mit Versionen von Microsoft Windows

Lokaler App-Zugriff und Microsoft Windows

Lokaler App-Zugriff verhält sich bei Microsoft Windows 8 und Windows Server 2012 anders als bei Windows 7 und Windows Server 2008 R2. Bei der Interaktion zwischen lokaler App-Zugriff und Windows tritt u. a. das folgende Verhalten auf.

- **Verhalten von Verknüpfungen in Windows 8 und Windows Server 2012**
 - Windows Store-Apps, die auf dem Client installiert sind, werden nicht als Teil der Verknüpfungen von lokalem App-Zugriff aufgelistet.
 - Bild- und Videodateien werden normalerweise standardmäßig mit Windows Store-Apps geöffnet. Lokaler App-Zugriff listet die Windows Store-Apps jedoch auf und öffnet Verknüpfungen mit Desktopanwendungen.
- **Local Programs**
 - In Windows 7 ist der Ordner im Startmenü verfügbar.
 - In Windows 8 ist der Ordner Local Programs nur verfügbar, wenn der Benutzer Alle Apps als Kategorie auf der Startseite auswählt. Nicht alle Unterordner werden in Local Programs angezeigt.
- **Windows 8-Grafikfunktionen für Anwendungen**
 - Desktopanwendungen sind auf den Desktopbereich beschränkt und werden von der Startseite bzw. Anwendungen im Windows 8-Stil vollständig abgedeckt.
 - Mit lokalem App-Zugriff verwendete Anwendungen verhalten sich jedoch bei der Verwendung von mehreren Monitoren nicht wie Desktopanwendungen. Bei der Verwendung mehrerer Monitore werden die Startseite und der Desktop auf unterschiedlichen Monitoren angezeigt.
- **Windows 8 und lokaler App-Zugriff mit URL-Umleitung**
 - **Windows 8:** Da bei Windows 8 Internet Explorer keine Add-Ons aktiviert sind, müssen Sie den Desktop-Internet Explorer zum Aktivieren von URL-Umleitung verwenden.
 - **Windows Server 2012:** In Windows Server 2012 werden Add-Ons von Internet Explorer aufgrund der verstärkten Sicherheitskonfiguration standardmäßig deaktiviert. So implementieren Sie URL-Umleitung:
 1. Deaktivieren Sie die verstärkte Sicherheitskonfiguration für Internet Explorer.
 2. Setzen Sie die Internet Explorer-Optionen zurück und starten Sie das Programm neu, um sicherzustellen, dass Add-Ons für Standardbenutzer aktiviert sind.

Taskleiste und Verknüpfungen

Mehrere Instanzen einer lokal ausgeführten Anwendung verhalten sich entsprechend den Taskleisteneinstellungen für den virtuellen Desktop. Allerdings gibt es bei einigen Verknüpfungen folgende Einschränkungen:

- Verknüpfungen zu lokal ausgeführten Anwendungen werden nicht mit ausgeführten Instanzen dieser Anwendungen gruppiert. Sie werden auch nicht mit ausgeführten Instanzen von gehosteten Anwendungen oder mit an gehosteten Anwendungen angehefteten Verknüpfungen gruppiert.

- Benutzer können Fenster von lokal ausgeführten Anwendungen nur von der Taskleiste aus schließen. Zwar können Benutzer die Fenster von lokalen Anwendungen in der Desktop-Taskleiste und im Startmenü anheften, jedoch starten die Anwendungen bei Verwendung dieser Verknüpfungen möglicherweise nicht konsistent.

Aktivieren von lokalem App-Zugriff auf Clientmaschinen mit Receiver

Zum Verwenden von lokaler App-Zugriff mit Citrix Receiver müssen folgende Voraussetzungen gegeben sein:

- Installieren Sie Receiver auf dem lokalen Clientcomputer.
- Führen Sie die folgenden Schritte aus, um die Konfiguration des lokalen App-Zugriffs abzuschließen:
 - — *Aktivieren von lokalem App-Zugriff während der Installation*
 - — *Aktivieren der Vorlage für den lokalen App-Zugriff mit dem Gruppenrichtlinien-Editor*
- Legen Sie die Richtlinieneinstellung für Lokalen App-Zugriff zulassen auf Aktiviert fest, wie unter — *Aktivieren von lokalem App-Zugriff in Studio* erläutert.

Aktivieren von lokalem App-Zugriff während der Installation

1. Navigieren Sie in der Windows-Eingabeaufforderung zu dem Verzeichnis, in dem Sie Citrix Receiver nach dem Download von der [Citrix.com-Downloadseite](https://www.citrix.com/downloads/citrix-receiver/citrix-receiver-windows.html) gespeichert haben.
2. Führen Sie den Citrix Receiver-Befehl aus, um Citrix Receiver zu installieren, wobei das Kennzeichen Allow_CLIENTHOSTEDAPPSURL wie in den folgenden Beispielen eingestellt sein muss:

```
CitrixReceiver.exe /ALLOW_CLIENTHOSTEDAPPSURL=1
```

Hierdurch wird der lokale App-Zugriff zusammen mit URL-Umleitung aktiviert und die Add-Ons für URL-Umleitung werden registriert.

Aktivieren der Vorlage für den lokalen App-Zugriff mit dem Gruppenrichtlinien-Editor

1. Öffnen Sie im Startmenü den Editor für lokale Gruppenrichtlinien, indem Sie den Befehl gpedit.msc im Feld "Ausführen" eingeben oder "Gruppenrichtlinie bearbeiten" suchen.
2. Fügen Sie die Vorlage icaclient.adm, die sich im Receiver-Konfigurationsordner befindet (normalerweise unter C:\Programme (x86)\Citrix\Online Plugin\Configuration), dem Editor für lokale Gruppenrichtlinien hinzu, indem Sie Computerkonfiguration auswählen.
3. Klicken Sie mit der rechten Maustaste auf Administrative Vorlagen und wählen Sie Vorlagen hinzufügen/entfernen > Hinzufügen.
4. Wenn Sie die Vorlage icaclient.adm hinzugefügt haben, erweitern Sie Administrative Vorlagen > Klassische administrative Vorlage (ADM) > Citrix Komponenten > Citrix Receiver > Benutzererfahrung.
Hinweis: Die Vorlage icaclient.adm ist auch in der Benutzerkonfiguration verfügbar, wenn Sie zur Computerkonfiguration hinzugefügt wurde.
5. Wählen Sie Einstellungen für lokaler App-Zugriff.
6. Wählen Sie Aktiviert und wählen Sie anschließend URL-Umleitung zulassen, um das Feature für die URL-Umleitung zu aktivieren. Registrieren Sie für die URL-Umleitung Browser-Add-Ons separat über die Befehlszeile.

Aktivieren von lokalem App-Zugriff in Studio

Wenn Sie die Einstellung für den lokalen App-Zugriff aktivieren, überlässt der VDA dem Client die Entscheidung darüber, ob Verknüpfungen für vom Administrator veröffentlichte Anwendungen oder Verknüpfungen für den lokalen App-Zugriff in der Sitzung aktiviert sind. Diese Richtlinie (wie auch die Richtlinie für die URL-Umleitung) gilt für den gesamten Computer. Weitere Informationen finden Sie unter [Einstellungen der Richtlinie "Lokaler App-Zugriff"](#).

1. Wählen Sie in Studio Richtlinie > Richtlinie bearbeiten.
2. Wählen Sie Lokalen App-Zugriff zulassen > Auswählen.
3. Wählen Sie Zugelassen > OK.
4. Klicken Sie auf Weiter und dann auf Fertig gestellt.

Verwenden von URL-Umleitung zum Starten von lokalen Anwendungen

In XenApp und XenDesktop verwenden gehostete Desktopsitzungen die URL-Umleitung zum Starten von lokalen Anwendungen. Durch URL-Umleitung wird die Anwendung unter mehr als einer URL-Adresse bereitgestellt. Durch Klicken auf eingebettete Links in einem Browser in einer Desktopsitzung wird ein lokaler Browser gestartet (basierend auf der URL-Sperrliste Ihres Browsers). Wenn Sie auf eine URL klicken, die nicht auf der Sperrliste steht, wird die URL erneut in der Sitzung geöffnet.

Zusätzlich zu URL-Umleitung können Sie auch FTA-Umleitung (File Type Association) verwenden. FTA startet lokale Anwendungen, wenn Dateien in einer Sitzung geöffnet werden sollen. Wenn die lokale Anwendung gestartet wird, muss sie Zugriff auf die Datei haben, um sie zu öffnen. Daher können Sie mit lokalen Anwendungen nur Dateien öffnen, die sich auf Netzwerkfreigaben oder auf Clientlaufwerken (mit CDM) befinden.

Wenn beispielsweise der PDF-Reader eine lokale Anwendung ist und die Datei \\client\C:\users\desktop\hugefile.pdf geöffnet werden soll, wird zum Öffnen der Datei der lokale PDF-Reader verwendet. Da die lokale Anwendung direkt auf die Datei zugreifen kann, erfolgt keine Netzwerkübertragung über ICA zum Öffnen der Datei.

Hinweis: Diese Features funktionieren nur in Desktopsitzungen und nicht in Anwendungssitzungen. Für Anwendungssitzungen können Sie nur Host-zu-Client-Inhaltsumleitung verwenden, wobei es sich um eine Art von Server FTA handelt. Diese FTA leitet bestimmte Protokolle an den Client um, z. B. HTTP, HTTPS, RTSP oder MMS. Wenn Sie beispielsweise nur eingebettete Links mit HTTP öffnen, werden die Links direkt in der Clientanwendung geöffnet. Für diese Art der Umleitung wird keine URL-Sperrliste oder URL-Positivliste unterstützt.

URL-Umleitung und lokaler App-Zugriff

Wenn "Lokaler App-Zugriff" für virtuelle Desktops aktiviert wurde, werden die URLs, die den Benutzern als Links von lokal ausgeführten Anwendungen, von benutzergehosteten Anwendungen oder als Verknüpfungen auf dem Desktop angezeigt werden, umgeleitet. Dabei wird eine der folgenden Methoden verwendet:

- Umleitung vom Computer des Benutzers zum gehosteten Desktop
- Umleitung vom XenApp- bzw. XenDesktop-Server auf den Computer des Benutzers
- Wiedergabe in der Umgebung, in der sie gestartet werden (keine Umleitung)

Bei der URL-Umleitung erfolgt eine URL-Abstimmung basierend auf einer vordefinierten Liste. Die URL wird am Endpunkt oder dem VDA-Browser selektiv gestartet. Verwenden Sie URL-Umleitung für Endbenutzer, die einen virtuellen Desktop als primären Arbeitsplatz nutzen.

Zur Angabe des Pfads für die Inhaltsumleitung von bestimmten Websites konfigurieren Sie die URL-Positivliste und die URL-Sperrliste auf dem Delivery Agent. Diese Listen bestehen aus mehrteiligen Registrierungsschlüsseln, die die Richtlinie für die URL-Umleitung identifizieren, wie unter [Einstellungen der Richtlinie "Lokaler App-Zugriff"](#) beschrieben.

Obwohl alle URLs auf VDA wiedergegeben werden können, gelten einige Ausnahmen:

- **Regions-/Gebietsschemainformationen:** Websites, die Gebietsschemainformationen benötigen, wie msn.com oder news.google.com (je nach Region wird eine bestimmte Seite geöffnet). Wenn VDA beispielsweise von einem Datenzentrum in Großbritannien bereitgestellt wird und der Client eine Verbindung aus Indien herstellt, würde der Benutzer erwarten, dass die Website in.msn.com erscheint, es wird jedoch uk.msn.com angezeigt.
- **Multimediainhalt:** Websites mit Rich-Media-Inhalt, die auf dem Clientgerät wiedergegeben werden, ermöglichen eine gewohnte Benutzererfahrung und das Einsparen von Bandbreite während die Funktionalität auch in Netzwerken mit hoher Latenz gewährleistet ist. Die Flash-Umleitung wird durch die Umleitung von Sites mit anderen Medientypen wie Silverlight ergänzt. Somit ist die Umgebung sehr sicher. Die vom Administrator genehmigten URLs werden auf dem Client ausgeführt, während die restlichen URLs an VDA weitergeleitet werden.

Aktivieren der URL-Umleitung bei der Installation von Receiver

Standardmäßig ist die URL-Umleitung auf Benutzergeräten deaktiviert, wenn Sie Receiver installieren. Sie können sie während der Installation über die Befehlszeile aktivieren. Gehen Sie wie folgt vor:

- Stellen Sie sicher, dass Sie Administratorrechte haben.
- Installieren Sie als Administrator Receiver für Alle Benutzer auf einem Computer.
Hinweis: Wenn Sie als Administrator eine Installation ausführen, ist der Installationspfad C:\Programme\Citrix\ICA Client. Überprüfen Sie den Installationspfad, damit sichergestellt ist, dass alle Benutzer Zugriff auf Receiver haben.
- Für Bereitstellungen mit dem Receiver Standardpaket geben Sie Folgendes ein:
CitrixReceiver.exe ALLOW_CLIENTHOSTEDAPPSURL=1

Hiermit werden die erforderlichen Browser-Add-Ons installiert und registriert sowie die für die Aktivierung des lokalen App-Zugriffs erforderlichen Client-Lockdown-Einstellungen einschließlich der URL-Umleitung aktiviert.

Registrieren von Browser-Add-Ons auf Geräten mit Receiver

Die URL-Umleitung verwendet Add-Ons für folgende Browser: Internet Explorer, Mozilla Firefox, Google Chrome und Firefox. Dieses Feature wird mit Receiver 4.0 installiert. Darüber hinaus können Sie Add-Ons auch über die folgenden Befehle registrieren:

- Internet Explorer
\\redirector.exe /regIE
- Firefox
\\redirector.exe /regFF
- Chrome
\\redirector.exe /regChrome
- Alle Browser
\\redirector.exe /regAll

Beispiel: Zum Registrieren von Add-Ons für Internet Explorer auf Receiver geben Sie Folgendes ein:

C:\Programme\Citrix\ICA Client\redirector.exe/regIE

Aufheben der Registrierung von Add-Ons

- Internet Explorer
\\redirector.exe /unregIE
- Firefox
\\redirector.exe /unregFF
- Chrome
\\redirector.exe /unregChrome

- Alle Browser
`\redirector.exe /unregAll`

Registrieren von Browser-Add-Ons auf gehosteten Desktops

Registrieren Sie Add-Ons auf gehosteten Desktops mit den folgenden Befehlen.

- Internet Explorer
`\VDARedirector.exe /regIE`
- Firefox
`\VDARedirector.exe /regFF`
- Chrome
`\VDARedirector.exe /regChrome`
- Alle Browser
`\VDARedirector.exe /regAll`

Beispiele

Registrieren Sie Add-Ons für Internet Explorer auf VDA für Desktopbetriebssysteme (Windows 7 oder Windows 8):

`C:\Programme\Citrix\ICAService\VDARedirector.exe /regIE`

Registrieren Sie Add-Ons für Internet Explorer auf einem VDA für Serverbetriebssysteme (Windows Server 2008 R2 oder Windows Server 2012):

`C:\Programme (x86)\Citrix\System32\VDARedirector.exe /regIE`

Aufheben der Registrierung von Add-Ons auf gehosteten Desktops

- Internet Explorer
`\VDARedirector.exe /unregIE`
- Firefox
`\VDARedirector.exe /unregFF`
- Chrome
`\VDARedirector.exe /unregChrome`
- Alle Browser
`\VDARedirector.exe /unregAll`

URL-Abfang in Browsern

Beschreibung	Konfiguration
Standardmäßig wird die eingegebene URL von Internet Explorer umgeleitet. Wenn die URL nicht in der Sperrliste enthalten ist sondern vom Browser oder der Website an eine andere URL-Adresse umgeleitet wird, wird die endgültige URL nicht umgeleitet, selbst wenn sie auf der Sperrliste steht.	Zum richtigen Funktionieren der URL-Umleitung müssen Sie bei entsprechender Aufforderung durch den Browser das Add-On aktivieren. Wenn die mit Internetoptionen verbundenen Add-Ons bzw. die die angeforderten Add-Ons deaktiviert sind, funktioniert die URL-Umleitung nicht richtig.
Firefox Add-Ons leiten URLs immer um.	Wenn ein Add-On von einem Benutzer oder von einem Installationsprogramm installiert wurde, bietet Firefox auf einer neuen Registerkarte die Möglichkeit, die Add-On-

Beschreibung	Konfiguration
	Installation zuzulassen oder zu stoppen. Sie müssen das Add-On zulassen, damit das Feature funktioniert.
Add-Ons von Google Chrome leiten die endgültige URL stets um, wenn es sich um geleitete und nicht eingegebene URLs handelt.	Die Erweiterungen wurden extern installiert. Wenn Sie die Erweiterung deaktivieren, funktioniert die URL-Umleitung in Google Chrome nicht. Wenn die URL-Umleitung im Inkognito-Modus erforderlich ist, lassen Sie durch Auswählen dieser Option zu, dass die Erweiterung im Inkognito-Modus ausgeführt wird.

Verbessern der Benutzererfahrung für Mobilgeräte

Sep 29, 2015

XenApp und XenDesktop bieten eine hervorragende Benutzererfahrung für mobile Benutzer, oftmals ohne dass eine spezielle Konfiguration erforderlich ist. Integrieren Sie XenApp und XenDesktop in anderen Citrix Produkten, um den mobilen Mitarbeitern zusätzliche Features zu bieten.

Feature:	Bereitgestellt von:
High Definition-Benutzererfahrung auf 3G- und 4G-Netzwerken	HDX In den meisten Fällen ergeben die standardmäßigen Citrix Richtlinienereinstellungen die beste Benutzererfahrung.
Für die Fingereingabe optimierte Schnittstelle zu virtuellen Desktops und Anwendungen, optimiert für Tablet-Geräte.	Einstellungen der Richtlinie "Mobilerfahrung" Die Standardeinstellungen liefern im Allgemeinen die beste Benutzererfahrung.
Remotezugriff auf Büro-PCs von anderen Geräten, einschließlich Smartphones, Tablets, Laptops und PCs	Remote-PC-Zugriff Citrix Receiver
Automatisch bereitgestellte Anwendungen für alle Benutzer eines Stores	Citrix StoreFront Citrix App Controller
E-Mail-basierte Kontenermittlung mit benutzerseitiger Kontoeinrichtung durch Eingabe der E-Mail-Adresse	Citrix StoreFront Citrix NetScaler Gateway (für Remoteverbindungen)
Sicherer Zugriff auf Stores, Desktops und Anwendungen	Citrix StoreFront Citrix NetScaler Gateway
Vereinfachte Druckverwaltung mit Netzwerkdrucken von jedem Gerät	Universeller Druckserver
Optimierte Leistung und Bereitstellung der Dienste für Zweigstellen und mobile Benutzer	Citrix CloudBridge
Rollenbasierte Verwaltung, Konfiguration und Sicherheit für Unternehmens- und mitarbeitereigene Mobilgeräte	Mobile Solutions Bundle (XenMobile MDM und CloudGateway)

Info zu Mobilitätsfeatures

Die Mobilitätsfeatures verbessern die Benutzererfahrung mit mobilen Geräten beim Zugreifen auf veröffentlichte Ressourcen. Zu den Features gehören u. a.:

- Verwenden der Steuerelemente des Mobilgeräts, statt nativer Windows-Steuerelemente wie Kombinationsfelder
- Automatische Anzeige der Gerätetastatur, wenn ein bearbeitbares Feld den Fokus erhält Bildlauf für Desktopsitzung

durchführen, um den Eingabebereich sichtbar zu machen

- Ein für die Fingereingabe optimierter Desktop für Mobilgeräte, der Folgendes bietet:
 - Verbesserter Zugriff auf das Windows-Startmenü: Tippen Sie auf die Schaltfläche "Start" und verwenden Sie die für die Fingereingabe optimierten Menüs für die Navigation auf Anwendungen und Dokumente. Starten Sie eine Anwendung oder öffnen Sie ein Dokument mit einem Tippen.
 - Mehrere Seiten mit Symbolen auf dem Desktop: Streifen Sie für die Navigation auf dem Desktop oder tippen Sie auf die Bildlaufsymbole.
 - Mit einmal Tippen zurück zum für die Fingereingabe optimierten Desktop, wenn er durch eine Anwendung im Vollbildmodus verdeckt ist: Tippen Sie auf das Symbol unten links auf dem Desktop.
 - Mit einmal Tippen zurück zum traditionellen Windows-Desktop: Tippen Sie auf das Symbol oben rechts auf dem Desktop, um zwischen dem für die Fingereingabe optimierten und Windows-Desktop hin- und herzuwechseln.
- Bereitstellen von Informationen zum Mobilgerätstandort (GPS) für Remoteanwendungssitzungen. Mit diesem Feature kann die Remoteanwendung Informationen zum Mobilgerätstandort von Citrix Receiver erhalten, sodass das Anwendungsverhalten so geändert werden kann, als ob die Anwendung lokal auf dem Mobilgerät ausgeführt wird.
- Das Mobile SDK für Windows-Apps ist eine Mobilgerätentwicklungsplattform, mit der Windows-Entwickler Anwendungen für Mobilgeräte in gängigen Programmiersprachen erstellen können. Das Mobile SDK für Windows-Apps enthält folgende Schnittstellen:
 - Steuern der Schaltflächenfunktionalität auf dem Mobilgerät
 - Einstellen der Bildschirmausrichtung
 - Aktivieren der Bildschirmtastatur
 - Verwenden lokaler Benutzeroberflächenelemente statt Windows-Steuerelemente
 - Zugriff auf Telefon-, SMS- und Kamerafunktionen des Geräts

Hinweis: In diesem Release werden die Audio- und Videoaufnahmefunktionen des neuesten Releases von Mobile SDK für Windows-Apps nicht unterstützt.

Behobene Probleme

- Die automatische Tastatur und das Kombinationsfeld auf dem Mobilgerät werden jetzt bei der zweiten Verwendung angezeigt. [253264]
- Der Richtlinienhilfetext für Tastatur automatisch anzeigen und Kombinationsfeld remoten ist richtig. [256356]
- Der Fehler "wfshell shell reagiert nicht mehr" wird nicht mehr angezeigt, wenn Sie Microsoft Editor starten, nach dem Sie das Programm beendet haben, bevor die Tastatur geöffnet wurde. [261973]

Bekannte Probleme

- Eine Anwendung mit einem .NET 4.0-Kalendersteuerelement kann abstürzen, wenn die Überwachung der Benutzeroberflächenautomatisierung von Microsoft in der gleichen Sitzung aktiv ist, und der Benutzer den Fokus auf das Kalendersteuerelement verschiebt. Dieses Problem entsteht aufgrund einer fehlenden Eigenschaft (ComponentResourceKey) im DataTemplate-Schlüssel für das .NET 4.0-Kalendersteuerelement. Eine Ressource, die auf der Designebene definiert ist, muss ComponentResourceKey als Schlüssel verwenden. Als Lösungsansatz für .NET 4.0 können Sie den DataTemplate-Schlüssel für das Kalendersteuerelement wie folgt einstellen:

Weitere Informationen finden Sie im Microsoft Support-Artikel [Null reference exception when running a .net app with UI automation](#). [261165]

- Das Feature "Automatische Tastatur" führt keinen Bildlauf der Anzeige aus, um den Eingabebereich anzuzeigen, wenn

die Auflösung auf einem iOS-Gerät auf einen anderen Wert als AutoAnpassen gesetzt ist. [267307]

- Bei einigen Vorgängen werden Anwendungen unerwartet zur Taskleiste des für die Fingereingabe optimierten Desktops minimiert. Tippen Sie auf das Taskleistensymbol, um die Anwendung wieder zu öffnen. [267606, 267609]
- Wenn eine Windows-Benachrichtigung angezeigt wird, wird die Windows-Taskleiste über der Taskleiste des für die Fingereingabe optimierten Desktops angezeigt. Schließen Sie die Benachrichtigung oder tippen Sie auf den Desktop, um die Taskleiste des für die Fingereingabe optimierten Desktops wieder anzuzeigen. [268911]
- Die Taskleiste des für die Fingereingabe optimierten Desktops wird mit dem Windows-Desktop angezeigt, wenn der Benutzer Alt-Tab drückt und auf das Zahnradsymbol vom Windows-Desktop tippt (Receiver für iOS). Tippen Sie auf das Symbol unten links, um die Anzeige zu berichtigen. [269535]
- Die Tastatur verdeckt den Eingabebereich, wenn die automatische Tastatur angezeigt wird, und ein Benutzer von Receiver für iOS das Gerät dreht. Der Benutzer kann die Anzeige verschieben oder das Gerät auf die Originalausrichtung drehen, um den Eingabebereich zu sehen. [269920]
- Die Taskleiste des für die Fingereingabe optimierten Desktops wird angezeigt, wenn eine Anwendung im Vollbildmodus ausgeführt wird. [272692]
- Die automatische Tastatur oder das native Kombinationsfeld des Geräts werden nicht für Anwendungen angezeigt, die mit erhöhten Rechten ausgeführt werden (Receiver für iOS). [273016]

Systemanforderungen für Mobilitätsfeatures

Devices

- Citrix Receiver für Android 3.x
- Citrix Receiver für iOS 5.5.x, 5.6.x, 5.7 und 5.8
- Citrix Receiver für Windows 8/RT 1.2 und 1.3

Anwendung

- Der Ortungsdienst wird für Anwendungen unterstützt, die das Windows 7-Location-API verwenden und Antworten basierend auf dem Clientortungssensor empfangen können.

Mobile SDK für Windows-Apps

- Entwicklungsbetriebssystem: Microsoft Windows 7 (x64) oder Windows 8
- Entwicklungsplattformen: Microsoft Visual Studio 2010 SP1 oder 2012
- Microsoft .NET Framework 3.5 SP1 und 4.0
- Microsoft Windows SDK 7.1 (für C++-Ortungsunterstützung)

Konfigurieren von Richtlinien für Mobilitätsfeatures

Die Einstellungen für Mobilitätsfeatures werden von den folgenden Einstellungen für die Citrix Benutzerkonfigurationsrichtlinien gesteuert.

Unter ICA > Mobilerfahrung:

- Automatische Anzeige der Tastatur
- Für Fingereingabe optimierten Desktop starten
- Kombinationsfelder remoten

Unter ICA > Clientsensoren > Standort:

- Anwendungen können den physischen Standort des Clientgeräts verwenden

Einstellen des Verhaltens für die Tastaturanzeige

Die Richtlinieneinstellung **Automatische Anzeige der Tastatur** legt das Verhalten der Tastatur in Anwendungssitzungen auf Mobilgeräten fest. Standardmäßig muss ein Benutzer einer Mobilgerätversion von Receiver die Tastatur manuell öffnen. Setzen Sie die Richtlinie auf Zulassen, damit die Tastatur automatisch geöffnet wird, wenn ein bearbeitbares Feld den Fokus hat. Wenn diese Einstellung zugelassen ist, kann ein Benutzer eine Sitzungseinstellung in Receiver für iOS ändern und verhindern, dass die Bildschirmtastatur automatisch angezeigt wird.

Bereitstellen einer für die Fingereingabe optimierten Oberfläche

Die Richtlinieneinstellung **Für die Fingereingabe optimierten Desktop starten** legt das Verhalten der Receiver-Oberfläche fest. In der Standardeinstellung wird für Tablets eine für die Fingereingabe optimierte Oberfläche verwendet. Stellen Sie diese Richtlinie auf Nicht zugelassen, um nur die Windows-Benutzeroberfläche zu verwenden.

Einstellen des Typs des angezeigten Kombinationsfelds

Die Richtlinieneinstellung Kombinationsfeld remoten legt den Typ des Kombinationsfeldes fest, das während Anwendungssitzungen auf Mobilgeräten angezeigt wird. Stellen Sie diese Richtlinie auf Zugelassen ein, um das geräteeigene Kombinationsfeld-Steurelement anzuzeigen. Wenn diese Einstellung zugelassen ist, kann ein Benutzer eine Sitzungseinstellung in Receiver für iOS ändern und das Windows-Kombinationsfeld verwenden.

Anwendungen können den Standort des Mobilgeräts verwenden

Mit der Richtlinieneinstellung Anwendung kann den Standort des Clientgeräts verwenden legen Sie fest, ob Anwendungen, die in einer XenApp-Sitzung auf einem Mobilgerät ausgeführt werden, den Standort des Clientgeräts verwenden können.

In der Standardeinstellung ist die Verwendung von Standortinformationen nicht zugelassen. Stellen Sie diese Richtlinie auf Zugelassen ein, um die Verwendung von Standortinformationen zu ermöglichen. Wenn diese Einstellung nicht zugelassen ist, kann ein Benutzer die Verwendung von Standortinformationen verhindern und eine Receiver-Anforderung für den Zugriff auf den Standort ablehnen. Android- und iOS-Geräte senden am Anfang jeder Sitzung eine Anforderung für die Standortinformationen.

Bereitstellen von Remote-PC-Zugriff für Benutzer

Sep 29, 2015

Über Remote-PC-Zugriff haben Desktopbenutzer sicheren Zugriff auf Ressourcen auf dem Büro-PC und genießen gleichzeitig die Vorteile der Citrix HDX-Technologie. Informationen zum Feature und Hinweise zu Überlegungen bei einer geplanten Bereitstellung finden Sie unter [Remote-PC-Zugriff](#).

Hinweis: Remote-PC-Zugriff gilt nur für XenDesktop-Lizenzen.

1. Zur Verwendung des Energieverwaltungsfeatures von Remote-PC-Zugriff (auch "Remote-PC-Zugriff-Wake-On-LAN" genannt), führen Sie die Konfigurationsaufgaben auf den PCs und in Microsoft System Center Configuration Manager (ConfigMgr) vor dem Erstellen der Remote-PC-Zugriff-Bereitstellung in Studio durch. Siehe [Microsoft System Center Configuration Manager und Remote-PC-Zugriff-Wake-On-LAN](#).
2. Beim Erstellen der ersten Remote-PC-Zugriff-Bereitstellung können Sie die Energieverwaltung für die Maschinen im Standardmaschinenkatalog für Remote-PC-Zugriff aktivieren oder deaktivieren. Wenn Sie die Energieverwaltung aktivieren, geben Sie die ConfigMgr-Verbindungsinformationen an. Geben Sie dann Benutzer und Computerkonten an. Weitere Informationen finden Sie unter [Erstellen einer Site](#). Die Erstellung einer Remote-PC-Bereitstellung schließt eine zukünftige VDI-Verwendung für die Site nicht aus.
Bei der Erstellung einer Remote-PC-Zugriff-Bereitstellung wird ein Standardmaschinenkatalog unter dem Namen
— *Remote-PC-Zugriff-Maschinen*
und eine Standardbereitstellungsgruppe unter dem Namen
— *Remote-PC-Zugriff-Desktops*
erstellt.
3. Beim Erstellen eines anderen Maschinenkatalogs für die Verwendung mit Remote-PC-Zugriff gilt Folgendes:
 - Betriebssystem: Wählen Sie Remote-PC-Zugriff und wählen Sie eine Energieverwaltungsverbindung aus. Das Verwenden der Energieverwaltung ist nicht obligatorisch. Wenn keine Energieverwaltungsverbindungen konfiguriert sind, können Sie nach dem Abschließen des Assistenten für die Erstellung eines Maschinenkatalogs eine erstellen (Verbindungstyp=Microsoft Configuration Manager-Wake-On-LAN) und dann den Maschinenkatalog unter Angabe der neuen Verbindung bearbeiten.
 - Maschinenkonten: Sie können angezeigte Maschinenkonten oder Organisationseinheiten wählen oder Maschinenkonten bzw. Organisationseinheiten hinzufügen.
4. Installieren Sie den VDA auf dem Büro-PC, der für lokalen und Remotezugriff verwendet wird. Normalerweise wird der VDA automatisch mit dem Verwaltungssoftwarepaket bereitgestellt. Bei Test- oder kleinen Bereitstellungen können Sie jedoch den VDA manuell auf jedem Büro-PC installieren.
Sobald der VDA installiert ist, wird der nächste Domänenbenutzer, der sich auf dem Büro-PC an einer Konsolensitzung anmeldet (lokal oder über RDP), automatisch dem Remote-PC-Desktop zugewiesen. Wenn in der Konsolensitzung weitere Domänenbenutzer angemeldet werden, werden sie der Desktopbenutzerliste unter Berücksichtigung der eingerichteten Beschränkungen ebenfalls hinzugefügt.
Hinweis: Zum Verwenden von RDP-Verbindungen außerhalb Ihrer XenApp- bzw. XenDesktop-Umgebung müssen Sie Benutzer oder Gruppen zur Gruppe der Benutzer mit direktem Zugriff hinzufügen.
5. Weisen Sie die Benutzer an, auf jedem Clientgerät, das sie für den Remotezugriff auf den Büro-PC verwenden, Citrix Receiver herunterzuladen und zu installieren. Citrix Receiver ist unter <http://www.citrix.com> oder in den Anwendungsverteilungssystemen für unterstützte Mobilgeräte verfügbar.

Sie können eine Energieverwaltungsverbindung zum Konfigurieren der erweiterten Einstellungen bearbeiten. Sie können Folgendes aktivieren:

- ConfigMgr-Aktivierungsproxy
- Wake-On-LAN-Pakete (Magic Packets). Wenn Sie Wake-On-LAN-Pakete aktivieren, können Sie eine Wake-On-LAN-

Übertragungsmethode auswählen: subnetzgesteuertes Broadcast oder Unicast.

Hinweis: Wake-On-LAN-Pakete werden vom Delivery Controller und nicht vom Microsoft System Center erstellt. Stellen Sie daher sicher, dass Firewalls und Routen dupliziert werden, sodass sie vorhandenen System Center-Regelsätzen entsprechen, damit die Pakete vom Controller durchkommen.

Der PC verwendet AMT-Energiebefehle (sofern unterstützt) und alle aktivierten erweiterten Einstellungen. Wenn der PC keine AMT-Befehle verwendet, werden die erweiterten Einstellungen verwendet.

Problembehandlung für Remote-PC-Zugriff

Der Delivery Controller schreibt die folgenden Diagnoseinformationen über Remote-PC-Zugriff in das Windows-Anwendungsereignisprotokoll. Informationsmeldungen werden nicht eingeschränkt. Fehlermeldungen werden durch Löschen doppelter Nachrichten eingeschränkt.

- 3300 (Informationsmeldung): Maschine zum Katalog hinzugefügt
- 3301 (Informationsmeldung): Maschine der Bereitstellungsgruppe hinzugefügt
- 3302 (Informationsmeldung): Maschine dem Benutzer zugewiesen
- 3303 (Fehler): Ausnahme

Wenn die Energieverwaltung aktiviert ist, können Maschinen, die sich in einem anderen Subnetz als der Controller befinden, ggf. nicht per subnetzgesteuertes Broadcast gestartet werden. Wenn Sie eine subnetzübergreifende Energieverwaltung mit subnetzgesteuertem Broadcast benötigen und AMT nicht unterstützt wird, versuchen Sie es mit dem Aktivierungsproxy oder Unicast (stellen Sie sicher, dass diese Einstellungen in den erweiterten Eigenschaften der Energieverwaltungsverbindung aktiviert sind).

Verhindern des Benutzerzugriffs auf den lokalen Desktop

Sep 29, 2015

Verwenden Sie Desktop Lock, wenn Benutzer nicht mit dem lokalen Desktop arbeiten müssen. In diesem Zugriffsszenario ersetzt der virtuelle Desktop den lokalen Desktop, sodass Benutzer mit dem virtuellen Desktop arbeiten, als wäre es der lokale.

Desktop Lock ist eine eigenständige Komponente, die mit XenApp, XenDesktop und Citrix VDI-in-a-Box auf den Markt gebracht wurde. Das Programm ist eine Alternative zu Desktop Viewer und wird hauptsächlich für umfunktionierte Windows-Computer und Thin Clients verwendet. Desktop Lock verwendet einen Ersatz für Windows Shell, in der kein Startmenü angezeigt wird, damit Benutzer nicht auf das lokale Betriebssystem zugreifen können. Außerdem wird ein Ersatz für den Windows Task-Manager verwendet.

Mit Desktop Lock können Benutzer auf Desktops von Bereitstellungsgruppen für Server- und Desktopbetriebssystemen zugreifen.

Systemanforderungen für Desktop Lock

Verwenden Sie Desktop Lock, wenn Benutzer nicht mit dem lokalen Desktop arbeiten müssen. In diesem Zugriffsszenario ersetzt der virtuelle Desktop den lokalen Desktop, sodass Benutzer mit dem virtuellen Desktop arbeiten, als wäre es der lokale.

Desktop Lock ist eine eigenständige Komponente, die mit XenApp, XenDesktop und Citrix VDI-in-a-Box auf den Markt gebracht wurde. Das Programm ist eine Alternative zu Desktop Viewer und wird hauptsächlich für umfunktionierte Windows-Computer und Thin Clients verwendet. Desktop Lock verwendet einen Ersatz für Windows Shell, in der kein Startmenü angezeigt wird, damit Benutzer nicht auf das lokale Betriebssystem zugreifen können. Außerdem wird ein Ersatz für den Windows Task-Manager verwendet.

Mit Desktop Lock können Benutzer auf Desktops von Bereitstellungsgruppen für Server- und Desktopbetriebssystemen zugreifen.

Betriebssystem

Desktop Lock wird auf den folgenden Benutzerbetriebssystemen unterstützt. Editionen oder Service Packs werden nur bei eingeschränkter Unterstützung aufgeführt:

- Windows 7, 32-Bit- und 64-Bit-Editionen (inkl. Embedded Edition)
- Windows XP Professional, 32-Bit- und 64-Bit-Edition
- Windows XP Embedded
- Windows Vista, 32-Bit- und 64-Bit-Editionen

Hinweis: Support für Windows XP endete zusammen mit dem erweiterten Support für Windows XP durch Microsoft am 8. April 2014. Der Support für Windows XP Embedded wird fortgesetzt.

Receiver

Citrix Receiver für Windows Enterprise 3.4-Paket

Umgebung

Benutzergeräte, auf denen Desktop Lock ausgeführt wird, werden nur unterstützt, wenn sie mit einem lokalen Netzwerk (LAN) verbunden sind.

In Domänen eingebundene und nicht in Domänen eingebundene Installationen

Sie können Desktop Lock auf Benutzergeräten installieren, die zu einer Domäne oder Arbeitsgruppe gehören. Das Systemverhalten und die Benutzererfahrung unterscheidet sich etwas auf den unterschiedlichen Gerätetypen.

In Installationen, die in Domänen eingebunden sind, wird das Windows-Benutzergerät Mitglied einer Active Directory-Domäne. Merkmale dieser Installationen:

- Benutzergeräte greifen über XenApp Services-URLs auf StoreFront-Stores zu. VDI-in-a-Box kann auch verwendet werden.
- Program Neighborhood Agent ersetzt die Windows Explorer-Shell und startet den ersten in alphabetischer Reihenfolge verfügbaren Desktop in einer zugewiesenen Desktopgruppe.
- Die Ersatzshell von Desktop Lock überschreibt nicht die Shell des Administrators.
- Windows Task-Manager wird durch eine Version ersetzt, mit der Benutzer den virtuellen Desktop neu starten können oder die Tastenkombination Strg+Alt+Entf an die Desktopsitzung übergeben wird.
- Auf Benutzergeräten unter Windows XP ruft die Tastenkombination Strg+Alt+Entf den Ersatz-Task-Manager auf.
- Das Benutzergerät imitiert den Grund für eine getrennte oder beendete Sitzung. Wenn der Benutzer zum Beispiel den virtuellen Desktop herunterfährt (und das Beenden der Sitzung bewirkt), wird das Gerät auch heruntergefahren.

Nicht in Domänen eingebundene Installationen werden hauptsächlich von Anbietern von Windows-Terminals verwendet, die kein Mitglied einer Active Directory-Domäne sind. Merkmale dieser Installationen:

- Benutzergeräte greifen auf StoreFront-Stores über Desktop Appliance-Sites zu.
- Benutzergeräte sind für die automatische Anmeldung konfiguriert und starten Internet Explorer im Kioskmodus. Die Domänenanmeldeseite wird nicht angezeigt.
- Der einzige Zugriffspunkt ist die Desktopgerätesite, die den ersten in alphabetischer Reihenfolge verfügbaren Desktop in einer zugewiesenen Bereitstellungsgruppe startet.
- Eine Version von Windows Task-Manager übergibt die Tastenkombination Strg+Alt+Entf an die Sitzungssitzung.

Unterstützung für mehrere Monitore

Desktop Lock unterstützt bis zu acht Monitore. Ein virtueller Desktop wird auf allen Monitoren angezeigt. Der primäre Monitor auf dem Gerät wird der primäre Monitor in der virtuellen Desktopsitzung.

Tastatureingabe in Desktop Lock-Sitzungen

In Desktop Lock-Sitzungen wird die Tastenkombination Windows-Logo-Taste + L an den lokalen Computer gesendet. In den meisten Fällen wird Strg + Alt + Entf an den lokalen

Computer gesendet. Umschalt + F2 kann nicht verwendet werden, um eine Vollbild-Desktopsitzung in den Fenstermodus zu schalten. Diese Tastenkombination kann jedoch in einer beliebigen Anwendungssitzung in der Desktopsitzung verwendet werden.

STRG + ESC und ALT + TAB werden an den virtuellen Remotedesktop gesendet.

Tastatureingaben, die die Einrastfunktion, die Anschlagverzögerung und Statusanzeige (Eingabehilfen von Microsoft) aktivieren, werden normalerweise an den lokalen Computer gesendet.

Hinweis: Bei Desktop Viewer werden einige Windows-Tastenkombination im Vergleich zu Desktop Lock anders behandelt. Informationen zu Desktop Viewer finden Sie in der Receiver-Dokumentation.

Installieren und Entfernen von Desktop Lock

Installieren von Desktop Lock

Mit diesen Schritten installieren Sie das Receiver für Windows-Plug-In, sodass virtuelle Desktops mit Desktop Lock angezeigt werden. Verwenden Sie diese Schritte nicht, wenn Desktop Viewer, Teil von Receiver, für die Benutzer verfügbar sein soll.

Wenn Sie Desktop Lock installieren, verwendet das System eine Ersatzshell. Um die Administration des Benutzergeräts nach der Installation zu ermöglichen, wird das Konto, das für die Installation von CitrixDesktopLock.msi verwendet wurde, beim Shellersatz ausgeschlossen. Wenn das Konto, das für die Installation von CitrixDesktopLock.msi verwendet wurde, später gelöscht wird, können Sie sich nicht am Gerät anmelden und es verwalten.

Sie sollten keine benutzerdefinierten Shells verwenden, da sie sich auf die für Desktop Lock-Sitzungen verwendete Ersatzshell negativ auswirken können.

Benutzer müssen für den Zugriff auf Desktops mit Desktop Lock Domänenkonten verwenden. Sie können keine lokalen Konten verwenden.

1. Melden Sie sich am Computer als lokaler Administrator an.
2. Führen Sie an einer Eingabeaufforderung CitrixReceiverEnterprise.exe mit der folgenden Syntax aus. Diese Datei ist im Ordner "Citrix Receiver" und Plug-Ins\Windows\Receiver auf dem Installationsmedium gespeichert:
`CitrixReceiverEnterprise.exe ADDLOCAL="ReceiverInside,ICA_Client,SSON,USB,DesktopViewer,Flash,PN_Agent,Vd3d" SERVER_LOCATION="my.server" ENABLE_SSON="Yes"`
Weitere Informationen zu den in diesem Befehl verwendeten Eigenschaften finden Sie im Abschnitt *— Konfigurieren und Installieren von Receiver für Windows mit Befehlszeilenparametern* in der Dokumentation für Receiver für Windows.
3. Geben Sie die URL der Services-Site ein, auf der sich die virtuellen Desktops befinden. Die URL muss im Format `http://Servername` oder `https://Servername` angegeben werden. Wenn Sie Hardware oder Software für das Load Balancing oder Failover verwenden, können Sie eine lastausgeglichene Adresse eingeben.
Wichtig: Stellen Sie sicher, dass die eingegebene URL richtig ist. Wenn die eingegebene URL falsch oder das Feld leer ist, und der Benutzer nach der Installation keine gültige URL eingibt, sind keine virtuellen oder lokalen Desktops verfügbar.
4. Navigieren Sie auf dem Installationsmedium zum Ordner "Citrix Receiver and Plug-Ins\Windows\Receiver" und doppelklicken Sie auf CitrixDesktopLock.msi. Der Assistent "Citrix Desktop Lock" wird angezeigt.
5. Lesen und akzeptieren Sie die Citrix Lizenzvereinbarung und klicken Sie auf Install. Der Installationsfortschritt wird angezeigt.
6. Klicken Sie im Dialogfeld Installation Completed auf Close.
7. Starten Sie das Benutzergerät neu, wenn Sie dazu aufgefordert werden. Wenn Sie Zugriffsrechte auf einen Desktop haben und Sie sich als ein Domänenbenutzer anmelden, wird das neu gestartete Gerät mit Desktop Lock angezeigt.

Entfernen von Desktop Lock

1. Melden Sie sich mit denselben Anmeldeinformationen des lokalen Administrators an, die bei der Installation von Desktop Lock verwendet wurden.
2. Öffnen Sie "Software" in der Systemsteuerung.
3. Entfernen Sie Citrix Desktop Lock.
4. Entfernen Sie Citrix Receiver oder Citrix Receiver (Enterprise).

Konfigurieren von Desktop Lock

Wichtig: Konfigurieren Sie Desktop Lock mit einem Administratorkonto. Dies muss das Konto sein, das Sie für die Installation verwendet haben.

Zum Konfigurieren von Desktop Lock mit der Gruppenrichtlinie laden Sie die folgenden ADM-Dateien unter Computerkonfiguration oder Benutzerkonfiguration > Administrative Vorlagen > Klassische administrative Vorlage (ADM) > Citrix Components:

- **icaclient.adm:** Informationen zum Erhalten dieser Datei finden Sie unter [Aktivieren der Smartcard Verwendung](#).
- **icaclient_usb.adm:** Sie finden die Datei im folgenden Ordner: %SystemDrive%\Programme\Citrix\ICA Client\Configuration\en.

Überlegungen bei der Konfiguration von Desktop Lock

Gewähren Sie pro Benutzer nur Zugriff auf einen virtuellen Desktop mit Desktop Lock.

Lassen Sie nicht zu, dass Benutzer virtuelle Desktops in den Ruhezustand setzen. Verhindern Sie dies mit Active Directory-Richtlinien.

Konfigurieren von USB-Einstellungen

Wenn ein Benutzer ein USB-Gerät anschließt, erfolgt ein automatisches Remoting des Geräts zum virtuellen Desktop. Benutzereingriffe sind nicht erforderlich. Der virtuelle Desktop steuert das USB-Gerät und zeigt es auf der Benutzeroberfläche an.

1. Aktivieren Sie die USB-Richtlinienregel, um die USB-Unterstützung in XenApp- bzw. XenDesktop-Bereitstellungen zu aktivieren. Weitere Informationen finden Sie unter [Einstellungen der Richtlinie "USB-Geräte"](#).
2. Aktivieren und konfigurieren Sie unter Citrix Receiver > Remoting client devices > Generic USB Remoting die Richtlinien Existing USB Devices, New USB Devices und USB Devices List In Desktop Viewer. Sie können die Richtlinie Show All Devices verwenden, um alle verbundenen USB-Geräte anzuzeigen, einschließlich solche, die den allgemeinen virtuellen Kanal für USB verwenden (z. B. Webcams und Memory Sticks).

Konfigurieren von Laufwerkszuordnung

Aktivieren und konfigurieren Sie unter Citrix Receiver > Remoting client devices die Richtlinie Client drive mapping.

Konfigurieren eines Mikrofons

Aktivieren und konfigurieren Sie unter Citrix Receiver > Remoting client devices die Richtlinie Client microphone.

Konfigurieren von Smartcards für die Verwendung mit Geräten mit Desktop Lock

Aktualisiert: 2015-03-05

Dieses Thema bietet einen Überblick über das Vorbereiten von Citrix StoreFront, Citrix Desktop Lock und Desktopgeräte für die Arbeit mit Smartcards.

Diese Anleitungen gelten für die mit Citrix Receiver für Windows Enterprise 3.4 verfügbare Version von Desktop Lock. Anleitungen zum Konfigurieren der mit Citrix 4.2 verfügbaren Version von Desktop Lock finden Sie unter [Receiver Desktop Lock](#).

1. Konfigurieren von StoreFront Weitere Informationen finden Sie unter [Installieren und Einrichten von StoreFront](#).
 1. Konfigurieren Sie den XML-Dienst zur Verwendung der DNS-Adressauflösung für Kerberos-Unterstützung.
 2. Konfigurieren Sie StoreFront-Sites für HTTPS-Zugriff, erstellen Sie ein Serverzertifikat, das von Ihrer Domänenzertifizierungsstelle signiert wurde und fügen Sie HTTPS-Bindung zur Standardwebsite hinzu.
 3. Stellen Sie sicher, dass Passthrough mit Smartcard aktiviert ist. Diese Option ist standardmäßig aktiviert.
 4. Aktivieren Sie Kerberos.
 5. Aktivieren Sie Kerberos und Passthrough mit Smartcard.
 6. Aktivieren Sie den anonymen Zugriff auf die IIS-Standardwebsite und verwenden Sie die integrierte Windows-Authentifizierung.
 7. Stellen Sie sicher, dass für die IIS-Standardwebsite kein SSL erforderlich ist, und dass Clientzertifikate ignoriert werden.
 8. Aktivieren Sie XenApp Services-Support.
2. Konfigurieren Sie lokale Computerrichtlinien auf dem Benutzergerät.
 1. Importieren Sie die icaclient.adm-Vorlage mit der Gruppenrichtlinien-Verwaltungskontrolle. Die Vorlage ist unter % Programme%\Citrix\ICA Client\Konfiguration\ verfügbar.
 2. Erweitern Sie Administrative Vorlagen> Klassische administrative Vorlage (ADM) > Citrix Komponenten > Citrix Receiver > Benutzerauthentifizierung.
 3. Aktivieren Sie Smartcard-Authentifizierung.
 4. Aktivieren Sie Lokaler Benutzername und Kennwort.
3. Konfigurieren Sie das Benutzergerät vor der Installation von Desktop Lock.
 1. Fügen Sie die URL für den Delivery Controller in der Windows Internet Explorer-Liste "Vertrauenswürdige Sites" hinzu.
 2. Fügen Sie die URL für die erste Desktopgruppe in der Windows Internet Explorer-Liste "Vertrauenswürdige Sites" hinzu. Verwenden Sie das folgende Format: desktop://desktop-20group-20name.
 3. Aktivieren Sie Internet Explorer für die automatische Anmeldung für vertrauenswürdige Sites.
4. Konfigurieren Sie das Benutzergerät nach der Installation von Desktop Lock.
 1. Bearbeiten Sie den Registrierungsschlüssel HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\PNAgent\ServerURL so, dass er auf PNAgent config.xml des Delivery Controllers verweist. Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab.

Wenn Sie Citrix Desktop Lock auf dem Benutzergerät installiert haben, wird eine konsistente Richtlinie für das Entfernen der Smartcard zwingend angewendet. Wird die Richtlinie für das Entfernen der Smartcard beispielsweise für den Desktop auf Abmelden erzwingen festgelegt, muss der Benutzer sich auch vom Benutzergerät abmelden, unabhängig davon, wie die Richtlinie dort eingestellt ist. Dadurch wird sichergestellt, dass das Benutzergerät sich nicht in einem inkonsistenten Zustand befindet. Dieses Verhalten gilt nur für Benutzergeräte mit Desktop Lock nicht Desktop Viewer.

Server-VDI

Sep 29, 2015

Mit dem Server-VDI-Feature (Virtual Desktop Infrastructure) können Sie einen Desktop von einem Serverbetriebssystem einem einzelnen Benutzer bereitstellen.

- Enterprise-Administratoren können Serverbetriebssysteme als VDI-Desktops bereitstellen. Dies ist für Benutzer, z. B. Techniker und Designer, nützlich.
- Dienstleister können Desktops von der Cloud anbieten. Diese Desktops entsprechen dem Microsoft Services Provider License Agreement (SPLA).

Mit der Citrix Richtlinieneinstellung "Enhanced Desktop Experience" können Sie das Serverbetriebssystem wie ein Windows 7-Betriebssystem aussehen lassen.

Die folgenden Features können nicht mit der Server-VDI verwendet werden:

- Personal vDisks
- HDX 3D Pro
- Gehostete Anwendungen
- Lokaler App-Zugriff
- Direkte (nicht vermittelte) Desktopverbindungen
- Remote-PC-Zugriff

Server-VDI wird auf den gleichen Serverbetriebssystemen wie der VDA für Windows Serverbetriebssysteme unterstützt, die in den Systemanforderungen aufgeführt sind.

Vorbereiten des Servers für die Installation

Stellen Sie auf dem Windows-Server sicher, dass die Rollendienste "Remotedesktopdienste" nicht installiert sind, und dass Benutzer auf eine Sitzung eingeschränkt sind:

- Stellen Sie mit dem Windows-Server-Manager sicher, dass die Rollendienste für Remotedesktopdienste nicht installiert sind. Wenn sie installiert sind, entfernen Sie die Dienste.
- Stellen Sie sicher, dass die Eigenschaft "Nur eine Sitzung pro Benutzer zulassen" aktiviert ist.
 - Unter Windows Server 2008 R2 greifen Sie auf diese Eigenschaft über Verwaltung > Remotedesktopdienste > Remotedesktop-Sitzungshostkonfiguration zu. Im Abschnitt Einstellungen bearbeiten > Allgemein muss die Einstellung Nur eine Sitzung pro Benutzer zulassen auf Ja festgelegt sein.
 - Unter Windows Server 2012 bearbeiten Sie die Registrierung und stellen Sie die Terminalservereinstellung ein. Legen Sie für den Registrierungsschlüssel HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TerminalServer den Eintrag DWORD fSingleSessionPerUser auf 1 fest.

Installieren Sie Microsoft .NET Framework 3.5 SP1 auf dem Server, bevor Sie den VDA installieren. (Die manuelle Installation dieser erforderlichen Software ist nur bei XenDesktop 7.0 bis 7.5 und XenApp 7.5 erforderlich.)

Installieren Sie den VDA und erstellen Sie einen Maschinenkatalog und eine Bereitstellungsgruppe

Installieren Sie über die Befehlszeilenschnittstelle des Installationsprogramms einen VDA für Windows-Desktopbetriebssysteme auf einem unterstützten Serverbetriebssystem. In der Standardeinstellung blockiert das Installationsprogramm den VDA für Windows-Desktopbetriebssysteme auf einem Serverbetriebssystem. Mit der Befehlszeile überschreiben Sie dieses Verhalten.

Erstellen Sie nach der VDA-Installation einen neuen dedizierten Maschinenkatalog für Server-VDI. Sie weisen den neuen

Maschinenkatalog dann einer neuen dedizierten Bereitstellungsgruppe zu.

1. Installieren Sie einen VDA über die Befehlszeilenschnittstelle mit den Optionen `/quiet` und `/servervdi` auf einem unterstützten Server oder einem Servermasterimage:

`XenDesktopVdaSetup.exe /quiet /servervdi`

Sie können den oder die Delivery Controller bei der VDA-Installation über die Befehlszeile mit der Option **`/controller`** festlegen.

Öffnen Sie mit der Option **`/enable_hdx_ports`** Ports in der Firewall, wenn die Firewall nicht manuell konfiguriert wird.

Schließen Sie keine Optionen für Features ein, die nicht mit Server-VDI unterstützt werden, u. a. `/baseimage`, `/enable_hdx_3d_pro` oder `/xa_server_location`.

Weitere Informationen finden Sie unter [Installieren an der Befehlszeile](#).

2. Geben Sie beim Erstellen eines Maschinenkatalogs für den Server, auf dem Sie den VDA installiert haben, Folgendes an:
 - Wählen Sie auf der Seite Betriebssystem und Hardware die Option Windows-Desktopbetriebssystem aus.
 - Fügen Sie auf der Seite Maschine den Server oder das Servermasterimage hinzu und wählen Sie dann den Server oder das Servermasterimage aus, auf dem Sie den VDA für Desktopbetriebssysteme installiert haben.
 - Geben Sie auf der Seite Zusammenfassung einen Maschinenkatalognamen und eine Beschreibung für Administratoren an, die ausdrücklich Server-VDI angeben. Dies ist die einzige Angabe in Studio, dass der Katalog Server-VDI unterstützt. Wenn Sie eine Suche in Studio durchführen, wird der erstellte Server-VDI-Katalog auf der Registerkarte Desktopbetriebssystemmaschinen angezeigt, obwohl der VDA auf einem Server installiert wurde.
3. Erstellen Sie eine Bereitstellungsgruppe und weisen Sie den Server-VDI-Katalog zu, den Sie im letzten Schritt erstellt haben.

Wenn Sie während der Installation des VDA keinen Delivery Controller festgelegt haben, geben sie ihn später über eine Citrix RichtlinienEinstellung, Active Directory oder durch Bearbeiten der Registrierungswerte des VDA-Computers an.

High Definition-Benutzererfahrung mit HDX

Sep 29, 2015

Citrix HDX enthält eine breite Palette an Technologien für eine High-Definition-Benutzererfahrung.

Auf dem Gerät:	HDX nutzt die Computingfähigkeiten der Benutzergeräte und verbessert und optimiert die Benutzererfahrung. Die HDX MediaStream-Technologie liefert eine gleichmäßigen, nahtlosen Multimediainhalt in den virtuellen Desktops oder Anwendungen der Benutzer. Mit Workspace Control können Benutzer virtuelle Desktops und Anwendungen anhalten und auf einem anderen Gerät an derselben Stelle weiterarbeiten.
Im Netzwerk:	HDX enthält erweiterte Optimierungs- und Beschleunigungsfunktionen und gewährleistet die beste Leistung in jedem Netzwerk, auch bei Verbindungen mit niedriger Bandbreite und bei WAN-Verbindungen mit hoher Latenz. HDX-Features passen sich den Änderungen in der Umgebung an und stimmen Lastausgleich und Bandbreite anhand optimaler Technologien für die jeweiligen Benutzerszenarios aufeinander ab, und zwar sowohl bei lokalem Zugriff auf die Desktops oder Anwendungen im Unternehmensnetzwerk als auch bei Remotezugriff von außerhalb des Unternehmens.
Im Datacenter:	HDX nutzt die Verarbeitungsleistung und die Skalierbarkeit von Servern für eine erweiterte Grafikleistung, unabhängig von den Funktionen des Clientgeräts. Komprimierte Multimediainformationen werden direkt im ursprünglichen Format an das Benutzergerät gesendet. Die in Citrix Director bereitgestellte HDX-Kanalüberwachung zeigt den Status der verbundenen HDX-Kanäle auf Benutzergeräten an. HDX Insight, die Integration von EdgeSight-Netzwerkinspektion und EdgeSight-Leistungsverwaltung mit Director, erfasst Daten zum ICA-Datenverkehr und bietet eine Dashboardansicht von Echtzeit- und historischen Informationen, wie die clientseitige und serverseitige ICA-Sitzungslatenz, die Bandbreitennutzung der ICA-Kanäle und die ICA-Roundtrip-Zeit für jede Sitzung.

Informationen zum Support und zu Systemanforderungen finden Sie unter [HDX-Systemanforderungen](#).

Die standardmäßig aktivierten Citrix Richtlinieneinstellungen liefern die beste Sofortlösung für die Mehrheit der Fälle. In den nachfolgenden Abschnitten wird beschrieben, wie Sie die Benutzererfahrung weiter optimieren, die Skalierbarkeit verbessern oder die Bandbreitenanforderungen reduzieren können. Weitere Informationen zum Arbeiten mit Citrix Richtlinien finden Sie unter [Verwalten von Citrix Richtlinien](#).

Hinweis: Sofern nicht anders in diesem Abschnitt angegeben stehen die HDX-Features für Windows-Serverbetriebssysteme, Windows-Desktopbetriebssysteme und Remote-PC-Zugriff-Desktops zur Verfügung.

Testen der HDX-Funktionen

1. Melden Sie sich an einem virtuellen Desktop an.
2. Um zu sehen, wie HDX hochwertige Videoinhalte auf virtuellen Desktops bereitstellt, rufen Sie eine Website mit HD-Videos auf, z. B. <http://www.microsoft.com/silverlight/iis-smooth-streaming/demo/>, und schauen Sie sich ein Video an.
3. Testen der Flash-Umleitung:
 1. Navigieren Sie zu <http://get.adobe.com/flashplayer/>, laden Sie Adobe Flash Player auf den virtuellen Desktop und das Benutzergerät herunter und installieren Sie die Anwendung.

2. Klicken Sie auf der Desktop Viewer-Symbolleiste auf Einstellungen. Klicken Sie im Dialogfeld Desktop Viewer-Einstellungen auf die Registerkarte Flash und wählen Sie Inhalt optimieren.
3. Um zu sehen, wie die Flash-Umleitung die Bereitstellung von Flash-Multimediainhalten auf virtuellen Desktops beschleunigt, gehen Sie zu einer Website mit Flash-Videos, z. B. <http://www.youtube.com/>, und sehen Sie sich ein Video auf dem Desktop an. Die Flash-Umleitung wird nahtlos integriert, sodass Benutzer nicht wissen, wann die Software ausgeführt wird. Sie können jedoch überprüfen, ob die Flash-Umleitung ausgeführt wird, indem Sie nach einem Farbblock Ausschau halten, der vorübergehend vor dem Start von Flash Player angezeigt wird.
4. Testen der HDX-Bereitstellung von High Definition-Audio für virtuelle Desktops:
 1. Konfigurieren Sie den Citrix Client für maximale Audioqualität. Weitere Informationen finden Sie in der Receiver-Dokumentation.
 2. Installieren Sie einen digitalen Audioplayer (z. B. iTunes über <http://www.apple.com/itunes/download/>) auf dem Desktop und spielen Sie einige Musikdateien ab.

HDX-Systemanforderungen

Sep 29, 2015

Anforderungen für die Desktopgestaltungsumleitung

Windows-Benutzergerät oder Thin Client:

- Unterstützung für DirectX 9
- Unterstützung für Pixel Shader 2.0 in der Hardware
- 32 Bit pro Pixel
- 1,5 GHz Prozessor (32-Bit oder 64-Bit)
- 1 GB RAM
- 128 MB Videospeicher auf der Grafikkarte oder dem integrierten Grafikprozessor

Hinweis: HDX überprüft das Windows-Gerät auf die erforderlichen GPU-Anforderungen und schaltet ggf. automatisch auf serverseitige Desktopgestaltung um. Windows-Geräte, die die erforderlichen GPU-Anforderungen erfüllen aber nicht die benötigte Prozessorgeschwindigkeit oder RAM-Spezifikationen haben, müssen in der GPO-Gruppe der Geräte erfasst werden, die von der Desktopgestaltungsumleitung ausgeschlossen sind.

Bandbreite:

- Mindestverfügbarkeit: 1,5 MBit/s
- Empfohlen: 5 MBit/s

Hinweis: Die Werte für die Mindest- und empfohlene Bandbreite schließen End-to-End-Latenz ein.

Anforderungen für die Bereitstellung von Windows Media

- Unterstützte Clients für Windows Media clientseitigen Inhaltsabruf, Windows Medienumleitung und Windows Media-Multimediatranscodierung in Echtzeit:
 - Receiver für Windows
 - Receiver für iOS
 - Receiver für Linux
- Zur Verwendung des clientseitigen Inhaltsabrufs von Windows Media auf Windows 8-Geräten legen Sie Citrix Multimedia Redirector als Standardprogramm fest: Navigieren Sie zu Systemsteuerung > Programme > Standardprogramme > Standardprogramme festlegen, wählen Sie Citrix Multimedia Redirector und klicken Sie auf Dieses Programm als Standard festlegen oder auf Standards für dieses Programm auswählen.
- Für GPU-Transcodierung: NVIDIA CUDA-fähige GPU mit Compute Capability 1.1 oder höher. Eine von NVIDIA-geführte Liste finden Sie unter <http://developer.nvidia.com/cuda/cuda-gpus>.

Anforderungen für die Flash-Umleitung

Benutzergerät:

- Zum Verwenden von Virtual Delivery Agent (VDA) aktivieren Sie eine Netzwerkverbindung zwischen dem Windows-Benutzergerät und VDA.
- Unterstützte Clients:
 - Receiver für Windows (für die zweite Generation der Flash-Umleitung)
 - Receiver für Linux (für die zweite Generation der Flash-Umleitung)
 - Citrix Online Plug-In 12.1 (nur für Legacy-Features der Flash-Umleitung)
- Unterstützte Adobe Flash Player:

- Windows: Die zweite Generation der Flash-Umleitung benötigt Adobe Flash Player - Other Browsers, das auch als NPAPI (Netscape Plugin Application Programming Interface) Flash Player bezeichnet wird.
- Linux: Die zweite Generation der Flash-Umleitung benötigt Adobe Flash Player for Other Linux oder Adobe Flash Player for Ubuntu.
- Für Legacy-Flash-Umleitung wird Adobe Flash Player für Windows Internet Explorer benötigt (auch ActiveX Player genannt)

Hinweis: Die Hauptversionsnummer von Flash Player auf dem Benutzergerät muss größer oder gleich der Hauptversionsnummer von Flash Player auf dem Server sein. Wenn eine ältere Version oder gar keine Version von Flash Player auf dem Benutzergerät installiert ist, werden Flash-Inhalte auf dem Server wiedergegeben.

Server oder Arbeitsstation mit Virtual Delivery Agents:

- Adobe Flash Player für Windows Internet Explorer (ActiveX Player)
- Internet Explorer Version 7, 8, 9, 10 (nicht im Metro-Modus)

Hinweis: Flash-Umleitung erfordert Internet Explorer auf dem Server; mit anderen Browsern werden Flash-Inhalte auf dem Server wiedergegeben.

- Geschützter Modus deaktiviert: Wählen Sie im Internet Explorer im Menü "Extras" die Option Internetoptionen aus. Deaktivieren Sie auf der Registerkarte "Sicherheit" das Kontrollkästchen Geschützten Modus aktivieren. Starten Sie Internet Explorer neu, damit die Änderungen wirksam werden.

HDX 3D Pro-Anforderungen

Server und Benutzergeräte müssen die hardwarebezogenen Mindestanforderungen für das installierte Betriebssystem erfüllen. Weitere Informationen über die Hardware, die Citrix zum Testen von HDX 3D Pro verwendet, finden Sie unter <http://support.citrix.com/article/CTX131385>.

Host:

Virtual Desktop Agent für HDX 3D Pro kann auf den folgenden Versionen von Windows installiert werden.

- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2
- Windows 8.1, 32-Bit- und 64-Bit-Editionen
- Windows 8, 32-Bit- und 64-Bit-Editionen
- Windows 7, 64-Bit-Editionen mit Service Pack 1
- Windows 7, 32-Bit-Editionen mit Service Pack 1
- Windows XP Professional, 64-Bit-Edition mit Service Pack 2
- Windows XP Professional, 32-Bit-Edition mit Service Pack 3

Der Virtual Delivery Agent für HDX 3D Pro kann mit XenApp 7.5, XenDesktop 7.5, XenDesktop 7.1, XenDesktop 7 oder XenDesktop 5.6 verwendet werden.

Sie können HDX 3D Pro verwenden, um beliebige Anwendungen bereitzustellen, die mit den unterstützten Hostbetriebssystemen kompatibel sind. Es eignet sich aber besonders für die Verwendung mit DirectX- und OpenGL-gesteuerten Anwendungen sowie mit reichhaltigen Medieninhalten wie z. B. Video.

Der Hostcomputer für die Anwendung kann entweder eine physische Maschine, eine VM mit GPU-Passthrough oder ein virtueller Grafikprozessor (vGPU) sein.

- GPU-Passthrough ist mit Citrix XenServer auf [Citrix Downloads](#) verfügbar. GPU-Passthrough ist auch mit VMware vSphere und VMware ESX verfügbar und wird in diesem Zusammenhang als vDGA (virtual Direct Graphics Acceleration) bezeichnet.

- Das vGPU-Feature ist mit Citrix XenServer unter [Citrix Virtual GPU Solution](#) verfügbar.

Als Minimalspezifikation für den Hostcomputer empfiehlt Citrix 4 GB RAM und vier virtuelle CPUs mit einer Taktfrequenz von 2,3 GHz oder höher.

Grafikprozessor (GPU):

- Im Hinblick auf CPU-basierte Komprimierung, einschließlich verlustfreier Komprimierung, unterstützt HDX 3D Pro alle Grafikkarten auf dem Hostcomputer, die mit der von Ihnen bereitgestellten Anwendung kompatibel sind.
- Für optimierten GPU-Framepufferzugriff mit NVIDIA GRID API erfordert HDX 3D Pro NVIDIA Quadro Smartcards mit den neuesten NVIDIA-Treibern. NVIDIA GRID liefert eine hohe Framerate und dadurch eine sehr interaktive Benutzererfahrung.
- Zu den Anforderungen für HDX 3D Pro zur Verwendung von vGPU mit XenServer gehören die Grafikkarten NVIDIA GRID K1 und K2. Umfassende Informationen zu vGPU-Anforderungen finden Sie unter [Citrix Virtual GPU Solution](#).

Benutzergerät:

Zum Zugriff auf Desktops oder Anwendungen, die mit XenDesktop und HDX 3D Pro bereitgestellt werden, müssen die Benutzer Citrix Receiver installieren. Weitere Informationen zu Systemanforderungen für Receiver finden Sie unter [Receiver und Plug-Ins](#).

- HDX 3D Pro unterstützt alle Monitoraufösungen, die von dem GPU auf dem Hostcomputer unterstützt werden. Um mit den empfohlenen Minimalspezifikationen für Benutzergeräte und GPUs eine optimale Leistung zu erzielen, empfiehlt Citrix jedoch eine maximale Monitorauflösung für Benutzergeräte von 1920 x 1200 Pixeln für LAN-Verbindungen sowie von 1280 x 1024 Pixeln für WAN-Verbindungen.
- Als Minimalspezifikation für Benutzergeräte empfiehlt Citrix mindestens 1 GB RAM und eine CPU mit einer Taktfrequenz von 1,6 GHz oder höher. Zur Verwendung des standardmäßigen Tiefenkomprimierungscodecs, der bei Verbindungen mit geringer Bandbreite erforderlich ist, ist eine leistungsfähigere CPU erforderlich, es sei denn, die Decodierung erfolgt in der Hardware. Für optimale Leistung empfiehlt Citrix die Ausstattung von Benutzergeräten mit mindestens 2 GB RAM und einer Dual-Core-CPU mit einer Taktfrequenz von 3 GHz oder höher.
- Bei Multi-Monitor-Zugriff empfiehlt Citrix Benutzergeräte mit Vierkern-CPU.

Benutzergeräte benötigen keinen dedizierten GPU für den Zugriff auf Desktops oder Anwendungen, die mit HDX 3D Pro bereitgestellt werden.

UDP-Audioanforderungen für Multistream-ICA

Unterstützter Client:

- Receiver für Windows
- Receiver für Linux 13

Anforderungen für Echounterdrückung

Unterstützter Client:

- Receiver für Windows

Anforderungen für Videokonferenzen mit Videokomprimierung

Unterstützte Clients:

- Receiver für Windows

- Receiver für Mac
- Receiver für Linux

Unterstützte Videokonferenzanwendungen:

- Citrix GoToMeeting HDFaces
- Adobe Connect
- Cisco WebEx
- IBM Sametime
- Microsoft Lync 2010
- Für Microsoft Office Communicator:
- Google+ Hangouts (kompatibel mit VDAs, auf denen Windows 7 ausgeführt wird)
- Skype 6.7: Zum Verwenden von Skype 6.7 über einen Windows-Client müssen Sie Änderungen an der Registrierung des Clients und des Servers vornehmen:
 - Navigieren Sie auf dem Client zum folgenden Registrierungsschlüssel:
HKEY_CURRENT_USER\Software\Citrix\HdxRealTime
 - Name: DefaultHeight

Typ: REG_DWORD

Wert: 240
 - Name: DefaultWidth

Typ: REG_DWORD

Wert: 320
 - Navigieren Sie auf dem Server zum folgenden Registrierungsschlüssel:
HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\Vd3d\Compatibility
 - Name: skype.exe

Typ: REG_DWORD

Wert: Stellen Sie 0 ein.

Andere Anforderungen an Benutzergeräte:

- Angemessene Hardware für die Audiowiedergabe
- DirectShow-kompatible Webcam (Webcam-Standardeinstellungen verwenden)
Hinweis: Bei Hardware-verschlüsselungsfähigen Webcams reduziert sich die clientseitige CPU-Nutzung.
- Installierte Webcam-Treiber (möglichst vom Kamera-Hersteller)

Optimieren der Grafik- und Multimediabereitstellung

Sep 29, 2015

XenApp und XenDesktop bieten standardmäßig hochwertige Grafiken und Videos, sodass für die meisten Benutzer keine Konfiguration erforderlich ist.

- HDX wählt automatisch die beste Bereitstellungsmethode basierend auf Client, Plattform, Anwendung und Bandbreite und nimmt dann selbständig entsprechend der geänderten Bedingungen eine Einstellung vor.
- HDX optimiert die Leistung von 2D- und 3D-Grafiken und Video.
- HDX liefert den Windows-Desktopbenutzern eine Windows Aero-Oberfläche auf jedem Client.
- HDX ermöglicht das Streamen von Multimediadateien für die Benutzergeräte direkt vom Quellenanbieter im Internet oder Intranet, ohne dass der Hostserver beteiligt wird. Wenn die Anforderungen für diesen clientseitigen Inhaltsabruf nicht erfüllt sind, erfolgt die Medienbereitstellung automatisch über Windows Media-Umleitung zur Wiedergabe von Medieninhalten auf dem Benutzergerät statt auf dem Hostserver. In den meisten Fällen ist keine Änderung der Richtlinien für Windows Media erforderlich.

Reduzieren der Bandbreite für Windows-Desktops

XenDesktop bietet standardmäßig eine hochgradig dynamische Windows Aero- oder Windows 8-Desktopdarstellung auf virtuellen Desktops von Windows-Benutzergeräten. Um dies zu gewährleisten, nutzt XenDesktop den Grafikprozessor (GPU) oder den integrierten Grafikprozessor (IGP) auf unterstützten Windows-Benutzergeräten für lokale DirectX Grafikwiedergabe. Dieses Feature wird als Desktopgestaltungsumleitung bezeichnet und gewährleistet hohe Skalierbarkeit auf dem Server.

Weitere Informationen finden Sie unter

— *What to do with all these choices*

unter <http://blogs.citrix.com/2013/11/06/go-supersonic-with-xendesktop-7-x-bandwidth-supercodecs/#Choices>.

Wichtig: Bei Nicht-Windows-Clients oder wenn die Desktopgestaltungsumleitung nicht verfügbar ist, verarbeitet HDX die Grafiken auf dem Server, komprimiert die Bitmapdatei und sendet sie über das Netzwerk an den Client. Aero-Designs werden unterstützt, sind jedoch nicht erforderlich.

Im neuesten Release von Citrix Receiver wurde die Leistung für die serverseitige Videowiedergabe verbessert.

Um die für Benutzersitzungen erforderliche Bandbreite zu reduzieren, sollten Sie die folgenden Richtlinieneinstellungen ändern. Beachten Sie, dass sich die Änderung der Einstellungen auf die Qualität der Benutzererfahrung auswirken kann.

- Desktopgestaltungsumleitung: gilt nur für Windows-Desktopbetriebssystemmaschinen, auf die über Windows-Benutzergeräte zugegriffen wird und nur für die Komposition des Windows-Desktops. Anwendungsfenster werden auf dem Server wiedergegeben, es sei denn, die Citrix Richtlinieneinstellung Zugriff auf lokale Anwendungen zulassen wurde auf Zugelassen eingestellt.
- Grafikqualität Desktopgestaltungsumleitung: Verwendet hochwertige Grafiken für die Desktopgestaltung, es sei denn, Seamless-Anwendungen oder Zugriff auf lokale Anwendungen ist aktiviert. Reduzieren Sie die Grafikqualität, um die Bandbreite zu reduzieren.
- Dynamische Fenstervorschau: Steuert die Anzeige von Seamlessfenstern, sodass folgende Vorschauoptionen auf den virtuellen Desktops von Benutzern angeboten werden:

Windows Aero-Vorschauoption	Beschreibung

Symbolleistenvorschau Windows Aero- Vorschauoption	Beschreibung Wenn der Benutzer auf das Symbol eines Fensters zeigt, wird ein Bild dieses Fensters über der Symbolleiste angezeigt.
Fenstervorschau	Wenn der Benutzer auf ein Symbolleistenvorschaubild zeigt, wird das Bild in voller Größe auf dem Bildschirm angezeigt.
Flip	Wenn der Benutzer Alt + Tab drückt, werden kleine Vorschausymbole für jedes geöffnete Fenster angezeigt.
Flip-3D	Wenn der Benutzer die Tabulator- und Windows-Tasten drückt, werden große Bilder der geöffneten Fenster überlappend auf dem Bildschirm angezeigt.

Zum Reduzieren der Bandbreite deaktivieren Sie diese Richtlinieneinstellung.

Verbessern der Bildqualität an Benutzergeräten

Die folgenden Richtlinieneinstellungen für "Visuelle Anzeige" steuern die Qualität der Bilder, die von virtuellen Desktops auf Benutzergeräte gesendet werden.

- **Bildqualität:** Steuert die optische Qualität der Bilder auf dem Benutzergerät.
 - **Mittel:** Mittlere Bildqualität bietet die beste Leistung und Bandbreiteneffizienz in den meisten Anwendungsfällen. Dies ist die Standardeinstellung.
 - **Hoch:** Die hohe Bildqualitätseinstellung wird empfohlen, wenn visuell verlustfreie Bildqualität gewünscht wird.
 - **Immer verlustfrei:** Mit der Einstellung "Immer verlustfrei" wird die pixelgenaue Bildverarbeitung sichergestellt. Diese Einstellung ist für besondere Anwendungsfälle erforderlich.
 - **Zu verlustfrei verbessern:** Mit der Einstellung "Zu verlustfrei verbessern" werden Anzeigedaten mit verlustreicher Komprimierung übertragen und nach der Übertragung wird die verlustfreie Bildqualität wieder hergestellt. Mit dieser Einstellung wird die Leistung bei Netzwerkverbindungen mit beschränkter Bandbreite verbessert.
- **Frameratesollwert:** Gibt die maximale Anzahl von Frames pro Sekunde an, die vom virtuellen Desktop zum Benutzergerät gesendet werden. Der Standardwert ist 30 Frames pro Sekunde. In vielen Fällen können Sie die Benutzererfahrung verbessern, indem Sie einen höheren Wert festlegen.

Wenn Benutzergeräte, z. B. Thin Clients, langsame CPUs haben, geben Sie einen niedrigeren Wert zum Verbessern der Benutzererfahrung ein.

Wenn die Skalierbarkeit des Servers ein Problem ist, geben Sie einen niedrigeren Wert an. Wenn die CPU-Auslastung des Servers bei 100 % oder knapp darunter liegt, denken Sie darüber nach, eine zusätzliche vCPU hinzuzufügen.

- **Anzeigespeicherlimit:** Gibt die maximale Größe des Videopuffers (in Kilobyte) für die Sitzung an. Der Standardwert ist 65536 KB. Für Verbindungen, die eine größere Farbtiefe und eine höhere Auflösung erfordern, erhöhen Sie den Grenzwert. Berechnen Sie den maximal erforderlichen Arbeitsspeicher mit dieser Formel:

$$(\text{Farbtiefe in Bits pro Pixel} / 8) * (\text{vertikale Auflösung in Pixel}) * (\text{horizontale Auflösung in Pixel}) = \text{erforderlicher Speicher in Byte}$$

Beispiel: Bei einer Farbtiefe von 32, einer vertikalen Auflösung von 600 und einer horizontalen Auflösung von 800 ergibt dies einen maximalen Arbeitsspeicher von $(32 \text{ bpp} / 8) * (600 \text{ Pixel}) * (800 \text{ Pixel}) = 1920000 \text{ Byte}$, also müssten Sie das Anzeigespeicherlimit auf 1920 KB einstellen.

Andere Farbtiefen als 32 Bit sind nur verfügbar, wenn die Richtlinie Legacygrafikmodus aktiviert ist.

Verbessern der Videokonferenzleistung

Zur Verbesserung der Bandbreiteneffizienz und Latenztoleranz bei Videokonferenzen in einer Sitzung wird standardmäßig HDX-Videokomprimierung für Webcams verwendet. Bei der HDX-Videokomprimierung werden die Webcamdaten über einen dedizierten virtuellen Multimediakanal gestreamt. Die HDX-Videokomprimierung beansprucht erheblich weniger Bandbreite im Vergleich zum isochronen HDX-Plug-n-Play und funktioniert gut über WAN-Verbindungen.

Allerdings können Receiver-Benutzer das Standardverhalten überschreiben, wenn sie in Desktop Viewer unter Mikrofon & Webcam die Einstellung Mikrofon und Webcam nicht verwenden auswählen. Um zu verhindern, dass Benutzer die HDX-Videokomprimierung ändern, deaktivieren Sie die Umleitung von USB-Geräten über die Richtlinienereinstellungen unter ICA> USB-Geräte.

Hinweis: Die HDX-Videokomprimierung ist bei Receiver unter Windows standardmäßig aktiviert, muss jedoch für Receiver unter Linux konfiguriert werden. Weitere Informationen finden Sie in der Receiver-Dokumentation in den eDocs. HDX-Videokomprimierung erfordert, dass die folgenden Richtlinienereinstellungen aktiviert sind. Alle diese Einstellungen sind standardmäßig aktiviert.

- Clientaudioumleitung
- Clientmikrofonumleitung
- Multimediakonferenzen
- Windows Media-Umleitung

Erzwingen von Softwarekomprimierung über Netzwerke mit geringer Bandbreite

Bei Webcams, die H.264-Hardwarecodierung unterstützen, verwendet HDX-Videokomprimierung die Hardwarecodierung standardmäßig. Die Hardware-Codierung beansprucht zusätzliche Bandbreite und ist nicht für Netzwerke mit geringer Bandbreite geeignet.

Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab. Zum Erzwingen der Softwarekomprimierung fügen Sie dem Registrierungsschlüssel HKCU\Software\Citrix\HdxRealTime den folgenden DWORD-Schlüsselwert hinzu: DeepCompress_ForceSWEncode = 1.

Bevorzugen der Skalierbarkeit des Servers vor der Benutzererfahrung

Bei Bereitstellungen bei denen die Skalierbarkeit des Servers wichtiger ist als die Benutzererfahrung, können Sie das Legacygrafiksystem verwenden, indem Sie die Einstellung Legacygrafikmodus einer Richtlinie hinzufügen und die einzelnen Richtlinienereinstellungen für Legacygrafiken konfigurieren. Ältere Grafiksysteme wirken sich besonders auf die Benutzererfahrung über WAN- und mobile Verbindungen aus.

Konfigurieren der Flash-Umleitung

Sep 29, 2015

Bei der Flash-Umleitung wird die Verarbeitung der meisten Adobe Flash-Inhalte (einschließlich Animationen, Videos und Anwendungen) an über LAN- und WAN-angeschlossene Windows-Benutzergeräte übertragen. Durch das Verschieben der Verarbeitung auf das Benutzergerät werden Serverressourcen gespart und die Server- und Netzwerklast reduziert. Dies führt zu größerer Skalierbarkeit, während gleichzeitig eine High Definition-Benutzererfahrung sichergestellt wird. Das Konfigurieren der Flash-Umleitung erfordert sowohl server- als auch clientseitige Einstellungen.

Achtung: Bei der Flash-Umleitung findet eine erhebliche Anzahl von Interaktionen zwischen dem Benutzergerät und den Serverkomponenten statt. Dieses Feature sollte daher nur in Umgebungen benutzt werden, in denen eine sicherheitsbedingte Trennung zwischen dem Benutzergerät und dem Server nicht erforderlich ist. Zudem müssen Benutzergeräten so konfiguriert werden, dass Sie dieses Feature nur mit vertrauenswürdigen Servern verwenden. Da für die Flash-Umleitung der Flash Player auf dem Benutzergerät installiert sein muss, ist diese Funktion nur zu aktivieren, wenn der Flash Player gesichert ist.

Die zweite Generation und Legacyversionen der Flash-Umleitung sind unabhängige Lösungen und werden in separaten virtuellen Kanälen ausgeführt.

- Legacy-Flash-Umleitungsfeatures werden nur auf dem Client unterstützt. Wenn eine ältere Version oder gar keine Version des Flash Players auf dem Benutzergerät installiert ist, werden Flash-Inhalte auf dem Server wiedergegeben.
- Die zweite Generation der Flash-Umleitung wird sowohl auf Clients als auch auf Servern unterstützt. Wenn der Client die Flash-Umleitung der zweiten Generation unterstützt, werden Flash-Inhalte auf dem Client wiedergegeben. Neue Features der zweiten Generation der Flash-Umleitung:
 - Unterstützung für Benutzerverbindungen über das WAN.
 - Intelligentes Fallback, wobei die Entscheidung für die Wiedergabe von Flash-Inhalten auf dem Server effizienzorientiert von Fall zu Fall getroffen wird.
 - Flash-URL-Kompatibilitätsliste, die festlegt, ob bestimmte URLs auf dem Client oder auf dem Server wiedergegeben oder blockiert werden.

Aktuelle Updates für HDX Flash-Kompatibilität finden Sie unter [CTX136588](#).

Flash-Ereignisprotokollierung

Flash-Umleitung verwendet die Windows-Ereignisprotokollierung auf dem Server, um Flash-Ereignisse zu protokollieren. Das Ereignisprotokoll gibt an, ob Flash-Umleitung verwendet wird und ob Probleme vorliegen. Folgendes gilt für alle Ereignisse, die von der Flash-Umleitung protokolliert werden:

- Die Flash-Umleitung meldet Ereignisse an das Anwendungsprotokoll.
- Unter Windows 8 und Windows 7 wird ein spezifisches Protokoll für die Flash-Umleitung im Knoten "Anwendungs- und Dienstprotokolle" angezeigt.
- Der Quellwert ist Flash.
- Es wird keine Kategorie angegeben.

Konfigurieren der Flash-Umleitung auf dem Server

Aktualisiert: 2014-8-11

Zum Konfigurieren der Flash-Umleitung auf dem Server, verwenden Sie die folgenden Citrix Richtlinieneinstellungen:

- Flash-Standardverhalten
- Flash - intelligentes Fallback
- URL-Liste für serverseitigen Flash-Inhaltsabruf
- Flash-URL-Kompatibilitätsliste
- Flash-Hintergrundfarbenliste

Festlegen des Flash-Standardverhaltens

Mit der Citrix Richtlinieneinstellung Flash-Standardverhalten können Sie das Standardverhalten der Flash-Beschleunigung festlegen. Standardmäßig ist die Flash-Umleitung aktiviert.

Zum Überschreiben dieses Standardverhaltens für einzelne Webseiten und Flash-Instanzen verwenden Sie die Einstellung Flash-URL-Kompatibilitätsliste.

Drei Optionen stehen zur Verfügung.

Option	Ergebnis
Flash Player blockieren	Flash-Umleitung sowie serverseitiges Rendering werden nicht verwendet. Der Benutzer kann keine Flash-Inhalte anzeigen.
Flash-Beschleunigung deaktivieren	Flash-Umleitung wird nicht verwendet. Der Benutzer kann auf dem Server wiedergegebene Flash-Inhalte anzeigen, wenn ein mit den Inhalten kompatibler — <i>Adobe Flash Player für Windows Internet Explorer</i> auf dem Server installiert ist.
Flash-Beschleunigung aktivieren	Die Flash-Umleitung wird verwendet. Dies ist die Standardeinstellung. Wichtig: Bei dieser Option ist es erforderlich, dass die Richtlinieneinstellung für Gruppenrichtlinienobjekte HDX MediaStream-Flash-Umleitung auf dem Benutzergerät aktivieren auf dem Benutzergerät aktiviert ist.

Steuern des intelligenten Fallbacks

"Flash - intelligentes Fallback" ist standardmäßig aktiviert, damit kleinere Flash-Filme (z. B. Werbefilme) erkannt und auf dem Server wiedergegeben werden und nicht auf das Benutzergerät umgeleitet werden.

"Flash - intelligentes Fallback" verursacht keine Unterbrechung oder Fehler beim Laden der Webseite oder der Flash-Anwendung.

Zum Umleiten aller Flash-Inhalte für die Wiedergabe auf dem Benutzergerät, deaktivieren Sie Flash - intelligentes Fallback.

Angabe von Websites für den serverseitigen Inhaltsabruf

Bei der standardmäßigen Flash-Umleitung werden Flash-Inhalte auf das Benutzergerät heruntergeladen und dort wiedergegeben. Mit der Einstellung für URL-Liste für serverseitigen Flash-Inhaltsabruf geben Sie die Websites an, deren Flash-Inhalte zum Server heruntergeladen und dann zum Benutzergerät gesendet werden. Diese Einstellung ist mit der Einstellung Serverseitigen Inhaltsabruf aktivieren auf dem Benutzergerät verknüpft und ist hauptsächlich für Intranetsites und interne Flash-Anwendungen vorgesehen. Sie funktioniert auch mit den meisten Websites und kann verwendet werden, wenn das Benutzergerät keinen direkten Internetzugang hat (z. B. wenn der XenApp- oder XenDesktop-Server die

Verbindung herstellt).

Hinweis: Beim serverseitigen Inhaltsabruf werden keine Flash-Anwendungen unterstützt, die Real Time Messaging Protocol (RTMP) verwenden; stattdessen wird serverseitiges Rendering verwendet, bei dem HTTP und HTTPS unterstützt werden. Zum Erstellen der Liste der zugelassenen URLs fügen Sie die Einstellung URL-Liste für serverseitigen Flash-Inhaltsabruf einer Richtlinie hinzu und klicken Sie auf Neu, um URLs der Liste hinzuzufügen. Gehen Sie beim Konfigurieren der URL-Liste für serverseitigen Flash-Inhaltsabruf wie folgt vor:

- Fügen Sie die URL der Flash-Anwendung hinzu und nicht die oberste Ebene der HTML-Seite, die den Flash Player der Liste hinzufügt.
- Verwenden Sie ein Sternchen am Anfang oder Ende der URL als Platzhalter, um die Liste zu erweitern.
- Verwenden Sie den Platzhalter am Ende, um alle untergeordneten URLs zuzulassen, z. B. http://www.zulässigeSite.com/*.
- Die Präfixe <http://> oder <https://> sind nicht erforderlich, werden aber verwendet, falls vorhanden.

Wichtig: Aktivieren Sie auf dem Benutzergerät die Richtlinieneinstellung Serverseitigen Inhaltsabruf aktivieren.

Festlegen der Wiedergabe von Flash-Inhalten

Über die Richtlinieneinstellung Flash-URL-Kompatibilitätsliste können Sie angeben, wie Flash-Inhalte von aufgelisteten Websites gehandhabt werden sollen:

- Wiedergabe auf dem Benutzergerät
- Wiedergabe auf dem Server
- Wiedergabe gesperrt

Gehen Sie beim Konfigurieren der Einstellung Flash-URL-Kompatibilitätsliste wie folgt vor:

- Sortieren Sie die Liste nach Priorität, sodass die wichtigsten URLs, Aktionen und Wiedergabeorte ganz oben stehen.
- Fügen Sie mit den Optionen Auf Server wiedergeben oder Blockieren Sites der Liste hinzu, deren Flash-Inhalte nicht korrekt auf dem Benutzergerät wiedergegeben werden.
- Verwenden Sie ein Sternchen am Anfang oder Ende der URL als Platzhalter, um die Liste zu erweitern.
- Verwenden Sie den Platzhalter am Ende, um alle untergeordneten URLs einzubeziehen (z. B. http://www.gesperrteSite.com/*).
- Die Präfixe <http://> oder <https://> sind nicht erforderlich, werden aber verwendet, falls vorhanden.

Hinzufügen der Einstellung Flash-URL-Kompatibilitätsliste zu einer Richtlinie

1. Klicken Sie auf Neu, um das Dialogfeld Flash-URL-Kompatibilitätslisteneintrag hinzufügen zu öffnen.
2. Wählen Sie eine Aktion: Auf Client wiedergeben, Auf Server wiedergeben oder Blockieren.
3. Geben Sie im Feld URL-Schema die URL der Website ein.
4. Wählen Sie die Flash-Instanz aus, die Sie als Auslöser festlegen möchten.
 - Bei Wahl von Jede wird die Aktion jedes Mal ausgeführt, wenn eine Flash-Instanz eine Verbindung mit der festgelegten Website herstellt.
 - Wählen Sie Bestimmte und geben Sie die Flash Player-ID ein, um den Vorgang nur ausführen zu lassen, wenn diese spezifische Flash-Instanz eine Verbindung mit der festgelegten Website herstellt.

Aktivieren der Farbabstimmung zwischen der Webseite und Flash-Instanzen

Um die Darstellung der Webseite bei Flash-Umleitung zu verbessern, verwenden Sie die Richtlinieneinstellung Flash-Hintergrundfarbenliste. Hiermit können Sie die Farben von Webseiten und Flash-Instanzen aufeinander abstimmen.

Gehen Sie beim Konfigurieren der Einstellung Flash-URL-Hintergrundfarbenliste wie folgt vor:

- Beste Ergebnisse erzielen Sie, wenn Sie eine Farbe verwenden, die normalerweise nicht auf der Webseite verwendet wird, z. B. Schwarz.

- Verwenden Sie einen Platzhalter am Ende, um die Anpassung für alle untergeordneten URLs zu erreichen, z. B. http://www.webseite.com/* FF0000).

Gehen Sie beim Hinzufügen der Einstellung Flash-URL-Hintergrundfarbenliste einer Richtlinie wie folgt vor:

1. Klicken Sie auf Neu und geben Sie die URL der Webseite gefolgt von der entsprechenden hexadezimalen Zahl der 24-Bit-Webfarbe ein (z. B. <http://www.sitetomatch.com/FF0000>).
2. Klicken Sie auf die Schaltfläche Annehmen.

Konfigurieren der Flash-Umleitung auf dem Benutzergerät

Zum Verwenden der Flash-Umleitung installieren Sie Citrix Receiver und Adobe Flash Player auf dem Benutzergerät. Es ist keine weitere Konfiguration auf dem Benutzergerät erforderlich. Sie können die Standardeinstellungen mit Gruppenrichtlinienobjekten wie in diesem Abschnitt beschrieben ändern.

Verwenden von Gruppenrichtlinienobjekten zum Konfigurieren der Flash-Umleitung auf dem Benutzergerät

1. Erstellen Sie ein Gruppenrichtlinienobjekt oder wählen Sie ein vorhandenes aus.
2. Importieren und fügen Sie die Datei HdxFlash-Client.adm (administrative Vorlage für HDX MediaStream Flash-Umleitung - Client) hinzu, die im folgenden Verzeichnis verfügbar ist:
 - **32-Bit-Computer:** %ProgramFiles%\Citrix\ICA Client\Configuration\Sprache
 - **64-Bit-Computer:** %Program Files (x86)%\Citrix\ICA Client\Configuration\Sprache

Die Einstellungen werden unter Administrative Vorlagen > Klassische administrative Vorlage (ADM) > HDX MediaStream Flash-Umleitung - Client aufgeführt.

Hinweis: Einzelheiten über das Erstellen von Gruppenrichtlinienobjekten und Importieren von Vorlagen finden Sie in der Dokumentation für Microsoft Active Directory unter <http://www.microsoft.com>.

Ändern der Verwendung der Flash-Umleitung

In Verbindung mit serverseitigen Einstellungen steuert die Richtlinie HDX MediaStream-Flash-Umleitung auf dem Benutzergerät, ob Adobe Flash-Inhalte für die lokale Wiedergabe auf dem Benutzergerät umgeleitet werden.

Hinweis: Wenn keine Konfiguration festgelegt ist und Desktop Lock verwendet wird, wird die Flash-Umleitung auf dem Benutzergerät standardmäßig aktiviert.

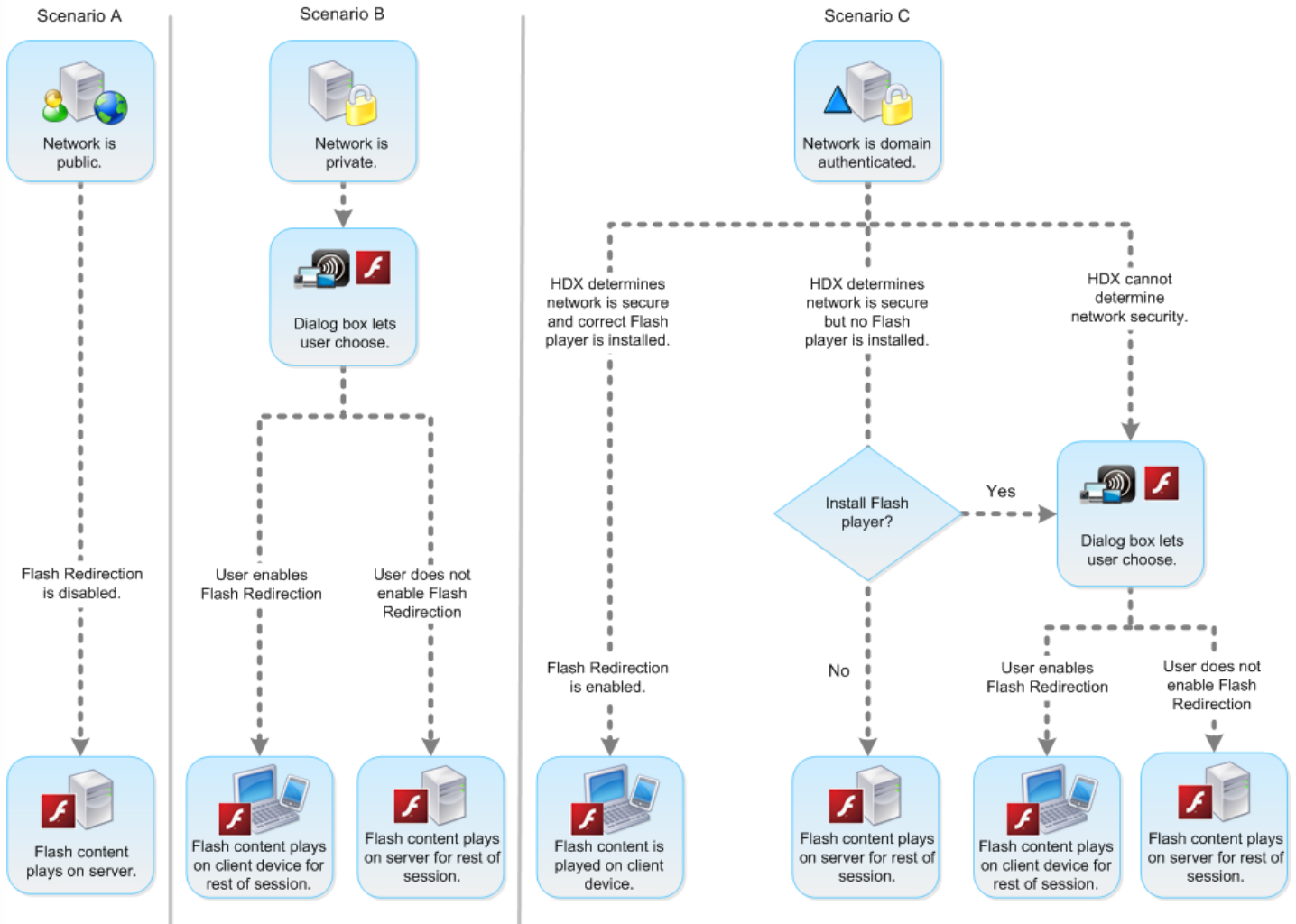
Standardmäßig ist die Flash-Umleitung aktiviert und ermittelt mit der intelligenten Netzwerkerkennung, wann Flash-Inhalte auf dem Benutzergerät wiedergegeben werden. Gehen Sie zum Ändern der Verwendung von Flash-Umleitung bzw. zum Deaktivieren der Flash-Umleitung auf dem Benutzergerät wie folgt vor:

1. Wählen Sie aus der Liste Einstellung den Eintrag HDX MediaStream Flash-Umleitung auf dem Benutzergerät aktivieren und klicken Sie auf Richtlinieneinstellung bearbeiten.
2. Wählen Sie Nicht konfiguriert, Aktiviert (Standardeinstellung) oder Deaktiviert.
3. Wenn Sie Aktiviert eingestellt haben, wählen Sie eine Option unter HDX MediaStream Flash-Umleitung verwenden aus:
 - Die Einstellung Nur zweite Generation dient dazu, die aktuelle Flash-Umleitung zu verwenden, wenn die erforderliche Konfiguration vorhanden ist bzw. auf serverseitige Wiedergabe umzuschalten, wenn dies nicht der Fall ist.
 - Bei Wahl von Immer wird die Flash-Umleitung immer verwendet. Flash-Inhalte werden auf dem Benutzergerät wiedergegeben.
 - Bei Wahl von Nie wird die Flash-Umleitung nie verwendet. Flash-Inhalte werden auf dem Server wiedergegeben.
 - Mit Fragen (Standardeinstellung) wird festgelegt, dass die Entscheidung, ob Flash-Umleitung verwendet wird, durch intelligente Netzwerkerkennung zur Beurteilung der Sicherheitsstufe des clientseitigen Netzwerks erfolgt. Wenn die Sicherheit des Netzwerks nicht bestimmt werden kann, erhält der Benutzer die Entscheidung über die Flash-Umleitung.

Es erscheint eine Aufforderung, die Flash-Umleitung zu aktivieren bzw. zu deaktivieren.

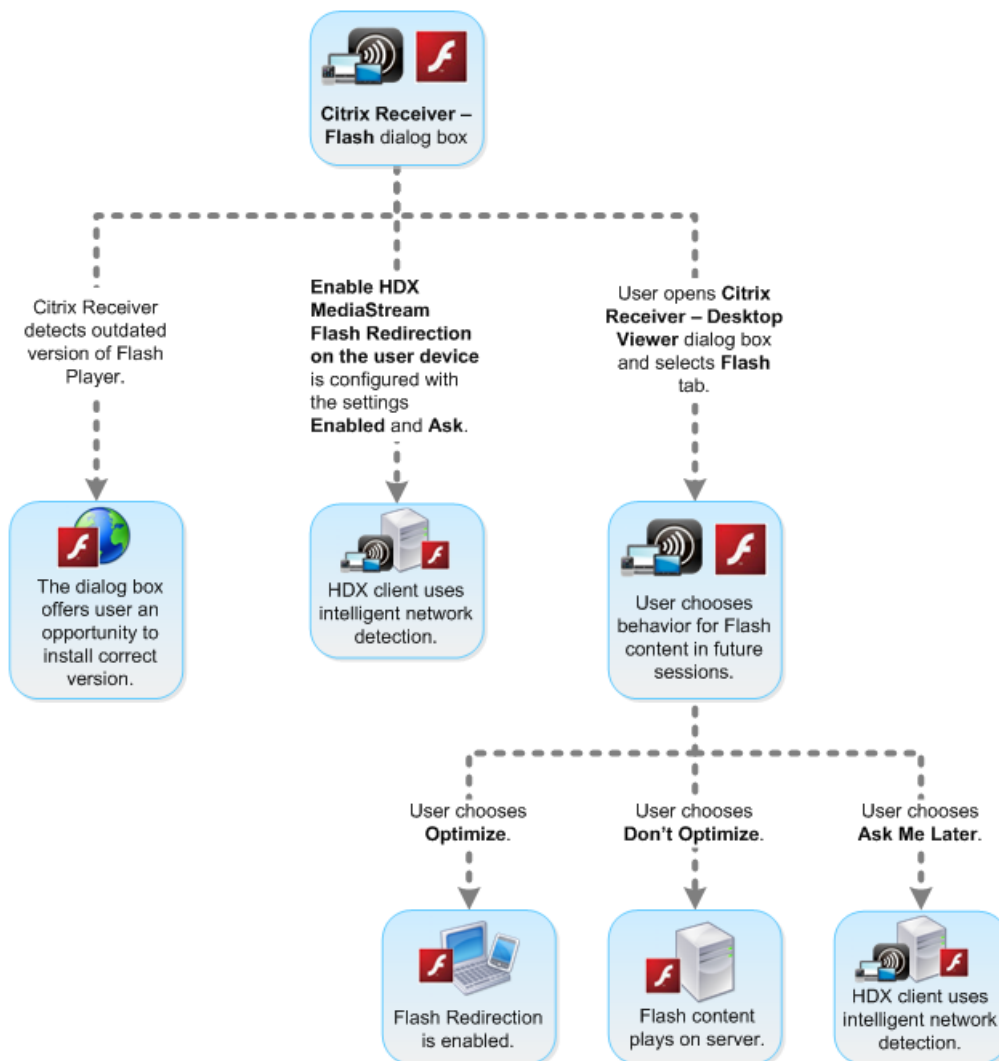
Die folgende Abbildung zeigt, wie die Flash-Umleitung für verschiedene Netzwerke verarbeitet wird.

Intelligent Network Detection for Flash Redirection



Hinweis: Benutzer können die intelligente Netzwerkerkennung über das Dialogfeld Citrix Receiver - Desktop Viewer-Einstellungen mit der Option Optimieren überschreiben oder über die Option Nicht optimieren auf der Registerkarte Flash. Welche Optionen zur Auswahl stehen, hängt davon ab, wie die Flash-Umleitung auf dem Benutzergerät konfiguriert ist (siehe folgende Abbildung).

User control of Flash redirection



Synchronisieren der clientseitigen HTTP-Cookies mit den serverseitigen HTTP-Cookies

Die Option zur serverseitigen HTTP-Cookie-Synchronisierung ist standardmäßig deaktiviert. Aktivieren Sie Funktion, um HTTP-Cookies vom Server herunterzuladen. Diese HTTP-Cookies werden dann für den clientseitigen Inhaltsabruf verwendet und können bei Bedarf von Sites mit Flash-Inhalten gelesen werden.

Hinweis: Clientseitige Cookies werden bei der Synchronisierung nicht ersetzt; sie sind weiterhin verfügbar, wenn die Synchronisierungsrichtlinie zu einem späteren Zeitpunkt deaktiviert wird.

1. Klicken Sie in der Liste Einstellung auf Synchronisierung der clientseitigen HTTP-Cookies mit den serverseitigen HTTP-Cookies aktivieren und klicken Sie auf Richtlinieneinstellung bearbeiten.
2. Wählen Sie Nicht konfiguriert, Aktiviert oder Deaktiviert (Standardeinstellung).

Aktivieren Sie den serverseitigen Inhaltsabruf.

Bei der standardmäßigen Flash-Umleitung werden Flash-Inhalte auf das Benutzergerät heruntergeladen und dort wiedergegeben. Bei Aktivierung des serverseitigen Inhaltsabrufs werden die Flash-Inhalte zum Server heruntergeladen und dann an das Benutzergerät gesendet. Die Inhalte werden auf dem Benutzergerät wiedergegeben, sofern keine überschreibende Richtlinie dies verhindert, z. B. wenn eine Site durch die Richtlinieneinstellung Flash-URL-Kompatibilitätsliste

blockiert wird.

Serverseitiger Inhaltsabruf wird in folgenden Fällen häufig verwendet:

- Das Benutzergerät stellt über Citrix NetScaler Gateway eine Verbindung zu internen Sites her.
- Das Benutzergerät hat keinen direkten Zugang zum Internet.

Hinweis: Der serverseitige Inhaltsabruf unterstützt keine Flash-Anwendungen, die Real Time Messaging Protocol (RTMP) verwenden. Stattdessen wird die serverseitige Wiedergabe für solche Sites verwendet.

Die zweite Generation der Flash-Umleitung unterstützt drei Optionen für den serverseitigen Inhaltsabruf (siehe folgende Tabelle). Zwei dieser Optionen umfassen das Zwischenspeichern serverseitiger Inhalte auf dem Benutzergerät; dies verbessert die Leistung, da wiederverwendeter Inhalt bereits wiedergabebereit auf dem Benutzergerät verfügbar ist.

Hinweis: Der Inhalt dieses Caches wird separat von anderen HTTP-Inhalten auf dem Benutzergerät gespeichert.

Option	Beschreibung
Deaktiviert	Deaktiviert serverseitigen Inhaltsabruf und überschreibt die Einstellung URL-Liste für serverseitigen Flash-Inhaltsabruf auf dem Server. Fallback auf serverseitigen Inhaltsabruf ist auch deaktiviert.
Aktiviert	Aktiviert serverseitigen Inhaltsabruf für Webseiten und Flash-Anwendungen, die in der URL-Liste für serverseitigen Flash-Inhaltsabruf angegeben sind. Fallback auf serverseitigen Inhaltsabruf ist verfügbar, aber der Inhalt wird nicht zwischengespeichert.
Aktiviert (permanentes Caching)	Aktiviert serverseitigen Inhaltsabruf für Webseiten und Flash-Anwendungen, die in der URL-Liste für serverseitigen Flash-Inhaltsabruf angegeben sind. Fallback auf serverseitigen Inhaltsabruf ist verfügbar. Über serverseitigen Inhaltsabruf erhaltener Inhalt wird auf dem Benutzergerät zwischengespeichert und von Sitzung zu Sitzung gespeichert.
Aktiviert (temporäres Caching)	Aktiviert serverseitigen Inhaltsabruf für Webseiten und Flash-Anwendungen, die in der URL-Liste für serverseitigen Flash-Inhaltsabruf angegeben sind. Fallback auf serverseitigen Inhaltsabruf ist verfügbar. Über serverseitigen Inhaltsabruf erhaltener Inhalt wird auf dem Benutzergerät zwischengespeichert und am Ende der Sitzung gelöscht.

Hinweis: Bei der Flash-Umleitung der zweiten Generation wird der Fallback auf serverseitigen Inhaltsabruf automatisch gestartet, wenn eine der oben genannten Optionen ausgewählt ist und der clientseitige Abruf von SWF-Dateien fehlschlägt.

Zum Aktivieren des serverseitigen Inhaltsabrufs müssen auf dem Clientgerät und dem Server folgende Einstellungen gegeben sein:

1. Wählen Sie in der Liste Einstellung den Eintrag Serverseitigen Inhaltsabruf aktivieren und klicken Sie auf Richtlinieneinstellung bearbeiten.
2. Wählen Sie Nicht konfiguriert, Aktiviert oder Deaktiviert (Standardeinstellung). Wenn Sie diese Einstellung aktivieren, wählen Sie eine der Optionen aus der Liste Serverseitiger Inhaltsabrufstatus:
 - Deaktiviert
Hinweis: Diese Einstellung wird in der Registrierung gespeichert.
 - Aktiviert
 - Aktiviert (permanentes Caching)
 - Aktiviert (temporäres Caching)
3. Aktivieren Sie auf dem Server die Richtlinieneinstellung URL-Liste für serverseitigen Flash-Inhaltsabruf und füllen Sie sie mit Ziel-URLs.

Umleiten der Benutzergeräte für clientseitigen Inhaltsabruf an andere Server

Sie können einen Versuch, Flash-Inhalte abzurufen, mit der Einstellung URL-Neuschreibregeln für clientseitigen Inhaltsabruf umleiten. Hierbei handelt es sich um ein Feature der Flash-Umleitung der zweiten Generation. Bei der Konfiguration dieses Features geben Sie mit regulären Ausdrücken in Perl zwei URL-Schemas an. Wenn das Benutzergerät versucht, Inhalte von einer Webseite abzurufen, die dem ersten Schema (dem

— *Übereinstimmungsschema*

) entspricht, werden sie an eine Webseite weitergeleitet, die mit dem zweiten Schema festgelegt wurde (dem

— *Ersetzungsschema*

).

Mit dieser Einstellung können Sie die Auswirkungen von Inhaltsübermittlungsnetzwerken (Content Delivery Networks, CDN) kompensieren. Manche Webseiten, die Flash-Inhalte bereitstellen, verwenden CDN-Umleitung, damit Benutzer den Inhalt vom nächstgelegenen Server in einer Gruppe von Servern, auf denen derselbe Inhalt zur Verfügung steht, abrufen können. Bei der Verwendung des clientseitigen Inhaltsabrufs der Flash-Umleitung wird der Flash-Inhalt vom Benutzergerät abgerufen, während der Rest der Webseite, auf der sich der Flash-Inhalt befindet, vom Server abgerufen wird. Wenn CDN verwendet wird, wird die Serveranfrage an den am nächsten gelegenen Server geleitet und die Benutzergeräatanfrage geht an denselben Standort. Beachten Sie, dass dies möglicherweise nicht der Speicherort ist, der dem Benutzergerät am nächsten liegt; je nach Entfernung, kann eine sichtbare Verzögerung zwischen dem Laden der Webseite und die Wiedergabe von Flash-Inhalten entstehen.

1. Wählen Sie in der Liste Einstellung den Eintrag URL-Neuschreibregeln für clientseitigen Inhaltsabruf und klicken Sie auf Richtlinieneinstellung bearbeiten.
2. Wählen Sie Nicht konfiguriert, Aktiviert oder Deaktiviert. Nicht konfiguriert ist der Standardwert; Deaktiviert bewirkt, dass alle URL-Neuschreibregeln des nächsten Schrittes ignoriert werden.
3. Wenn Sie Aktiviert ausgewählt haben, klicken Sie auf Anzeigen. Verwenden Sie die Perl-Syntax für reguläre Ausdrücke und geben Sie das URL-Schema im Feld Wert und das neu geschriebene URL-Format im Feld Wert ein.

HDX 3D Pro

Sep 29, 2015

Mit HDX 3D Pro können Sie Desktops und Anwendungen bereitstellen, die am besten mit einem Grafikprozessor (GPU) für die Hardware-Beschleunigung arbeiten, einschließlich professioneller auf OpenGL und DirectX basierender 3D-Grafikanwendungen. (Der Standard-VDA unterstützt die GPU-Beschleunigung nur für DirectX.)

Beispiele für professionelle 3D-Anwendungen:

- CAD-, CAM- und CAE-Anwendungen
- Geografische Informationssystemsoftware (GIS)
- Bildarchivierungskommunikationssystem (PACS) für bildgebende Diagnostik
- Anwendungen, die die aktuellen Versionen von OpenGL, DirectX, NVidia CUDA und OpenCL verwenden
- Rechenintensive Nichtgrafik-Anwendungen, die NVIDIA CUDA-GPUs (Compute Unified Device Architecture) für paralleles Computing verwenden

HDX 3D Pro bietet die beste bandbreitenunabhängige Benutzererfahrung:

- Für WAN-Verbindungen: Stellen Sie eine interaktive Benutzererfahrung über WAN-Verbindungen mit geringen Bandbreiten bis zu 1,5 MBit/s bereit.
- LAN-Verbindungen: Stellen Sie eine Benutzererfahrung wie bei einem lokalen Desktop bei LAN-Verbindungen mit Bandbreiten von 100 MBit/s bereit.

Mit HDX 3D Pro können Sie komplexe und teure Arbeitsstationen durch einfache Benutzergeräte ersetzen, da die Grafikverarbeitung in das Datenzentrum für eine zentralisierte Verwaltung verschoben wird.

Hinweis: Mit dem HDX Monitor-Tool, das das Tool für die Systemdiagnose ersetzt, können Sie den Betrieb und die Konfiguration von HDX-Visualisierungstechnologien überprüfen und HDX-Probleme diagnostizieren und beheben. Das Tool und weitere Informationen stehen unter <https://taas.citrix.com/hdx/download> zur Verfügung.

HDX 3D Pro stellt Windows-Desktopbetriebssystemmaschinen und Windows-Serverbetriebssystemmaschinen die GPU-Beschleunigung bereit. HDX 3D Pro bietet vGPU-Beschleunigung für Windows-Desktopbetriebssystemmaschinen bei Verwendung mit Citrix XenServer und NVIDIA GRID-GPUs. Informationen zu den unterstützten XenServer-Versionen finden Sie unter [Citrix Virtual GPU Solution](#).

GPU-Beschleunigung für Windows Desktopbetriebssysteme

Feb 17, 2016

Mit HDX 3D Pro können Sie grafikintensive Anwendungen gemäß den Anforderungen der Benutzer als Teil gehosteter Desktops oder Anwendungen auf Desktopbetriebssystemmaschinen bereitstellen. HDX 3D Pro unterstützt physische Hostcomputer (einschließlich Desktop-, Blade- und Rack-Arbeitsstationen), XenServer-VMs mit GPU-Passthrough und XenServer-VMs mit virtuellem GPU (vGPU).

Mit dem XenServer-Feature GPU-Passthrough können Sie VMs mit exklusivem Zugriff auf dedizierte Hardware für die Grafikverarbeitung erstellen. Sie können mehrere GPUs auf dem Hypervisor installieren und VMs jeder dieser GPUs einzeln zuweisen.

Mit dem vGPU-Feature von XenServer können mehrere virtuelle Maschinen die Grafikverarbeitungsleistung eines einzelnen GPU direkt nutzen.

Neue Funktionen in diesem Release

- **vGPU:** Mit dem vGPU-Feature (virtueller Grafikprozessor) können mehrere virtuelle Maschinen die Grafikverarbeitungsleistung eines einzelnen physischen GPU direkt nutzen. Sie können den hardwarebeschleunigten vGPU-Zugriff für Windows-Desktop-VDI-Arbeitslasten verwenden. Die echte gemeinsame Hardware-GPU-Nutzung bietet vollständige Windows 7- oder Windows 2008 R2 SP1-Desktops, die für Benutzer mit komplexen und anspruchsvollen Designanforderungen geeignet sind. Die gemeinsame GPU-Nutzung wird für NVIDIA GRID K1- und K2-Karten unterstützt und verwendet die gleichen NVIDIA-Grafiktreiber, die auf nicht-virtualisierten Betriebssystemen bereitgestellt werden.
- **Unterstützung für Flash-Umleitung auf mehr Betriebssystemen:** Flash-Umleitung wird jetzt für Windows Server 2012 R2, Windows Server 2012, Windows 8.1 und Windows 8 VDAs unterstützt. Für diese Betriebssysteme ist Flash-Umleitung standardmäßig aktiviert.
- **Positionen der Anwendungsfenster werden in lokalen Sitzungen beibehalten:** Wenn ein Benutzer von einer lokalen Sitzung zu einer Remotesitzung wechselt und dann wieder zur lokalen Sitzung zurückkehrt, werden die Anwendungsfenster wieder auf ihre vorherigen Positionen in der lokalen Sitzung zurückgesetzt. Fensterpositionen werden nach Clientnamen gespeichert.

Hauptfeatures

- **Adaptive, auf dem H.264-Standard basierende Tiefenkomprimierung für optimale Leistung bei WAN-Verbindungen und drahtlosen Verbindungen:** HDX 3D Pro verwendet die CPU-basierte Tiefenkomprimierung als Standardkomprimierungsverfahren für die Verschlüsselung. Dadurch wird eine optimale Komprimierung gewährleistet, die sich dynamisch an die Netzwerkbedingungen anpasst.
Der auf dem H.264-Standard basierende Tiefenkomprimierungscodec konkurriert nicht mehr mit der Grafikwiedergabe um CUDA-Kerne auf der NVIDIA-GPU. Der Tiefenkomprimierungscodec wird auf der CPU ausgeführt und bietet überragende Bandbreiteneffizienz im Vergleich zu vorherigen HDX 3D Pro-Releases.
- **Verlustfreie Komprimierung für besondere Anwendungsfälle:** HDX 3D Pro bietet einen verlustfreien CPU-basierten Codec zur Unterstützung von Anwendungen, in denen pixelgenaue Grafiken unerlässlich sind, z. B. für die bildgebende Diagnostik. Der auf dem H.264-Standard basierende Tiefenkomprimierungscodec bietet eine weitaus bessere Bandbreiteneffizienz als frühere XenDesktop-Releases. Verlustfreie Komprimierung wird nur für besondere Anwendungsfälle empfohlen, da sie wesentlich mehr Netzwerk- und Verarbeitungsressourcen benötigt.
Bei Verwendung von verlustfreier Komprimierung:
 - Die Anzeige für Verlustfreiheit, ein Symbol in der Taskleiste, zeigt an, ob es sich bei der Bildschirmanzeige um einen verlustreichen oder verlustfreien Frame handelt. Dies ist nützlich, wenn VisualQuality auf BuildToLossless eingestellt ist. Die Anzeige für Verlustfreiheit wird grün, wenn die gesendeten Frames verlustfrei sind.
 - Über die Umschaltung für Verlustfreiheit können die Benutzer jederzeit innerhalb der Sitzung in den immer verlustfreien Modus wechseln. Zum Aktivieren

oder Deaktivieren des verlustfreien Modus in einer Sitzung können Sie jederzeit mit der rechten Maustaste auf das Symbol klicken oder verwenden Sie die Tastenkombination ALT + UMSCHALT + 1.

Für verlustfreie Komprimierung: HDX 3D Pro verwendet den verlustfreien Codec für die Komprimierung unabhängig von dem durch die Richtlinie ausgewählten Codec.

Für die verlustreiche Komprimierung: HDX 3D Pro verwendet den ursprünglichen Codec, entweder den Standard oder den über die Richtlinie ausgewählten Codec.

Hinweis: Einstellungen für die Umschaltung für Verlustfreiheit werden nicht für zukünftige Sitzungen gespeichert. Wenn Sie für alle Verbindungen den verlustfreien Codec verwenden möchten, legen Sie für die Richtlinie Bildqualität Immer verlustfrei fest.

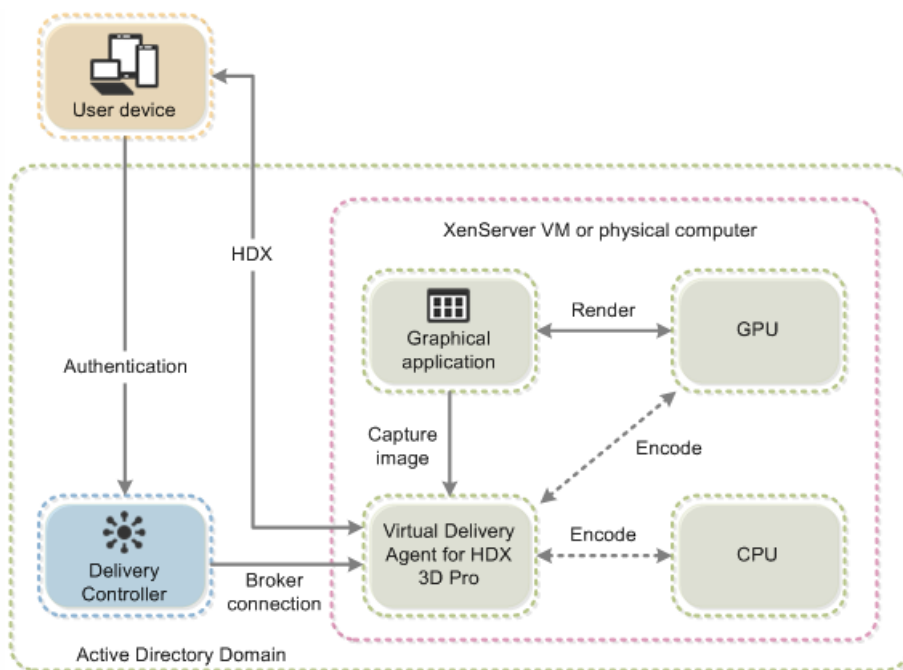
- **Unterstützung mehrerer Monitore:** Für Windows 7- und Windows 8-Desktops unterstützt HDX 3D Pro Benutzergeräte mit bis zu vier Monitoren. Benutzer können ihre Monitore beliebig konfigurieren sowie Monitore mit unterschiedlichen Auflösungen und Ausrichtungen kombinieren. Die Anzahl der Monitore wird nur durch die Leistungsfähigkeit des GPU auf dem Hostcomputer, des Benutzergeräts und der verfügbaren Bandbreite begrenzt. Auf Windows XP-Desktops bietet HDX 3D Pro eingeschränkte Unterstützung für Dual-Monitor-Zugriff. Der XenDesktop 5.6-VDA in XenDesktop 7, XenDesktop 7.1, XenDesktop 7.5 und XenApp 7.5 ist für die Bereitstellung virtueller Windows XP-Desktops und Anwendungen erforderlich.
- **Unterstützung für hochauflösende Monitore:** HDX 3D Pro unterstützt alle Monitorauflösungen und ist nur durch die Leistungsfähigkeit des GPU auf dem Hostcomputer eingeschränkt.
- **Dynamische Auflösung:** In diesem Release können Sie das Fenster des virtuellen Desktops oder der Anwendung auf eine beliebige Auflösung einstellen. **Hinweis:** Die einzige unterstützte Methode zum Ändern der Auflösung ist das Anpassen des VDA-Sitzungsfensters. Das Ändern der Auflösung in der VDA-Sitzung (über "Systemsteuerung" > "Darstellung und Anpassung" > "Anzeige" > "Bildschirmauflösung") wird nicht unterstützt.
- **Unterstützung für die NVIDIA-Kepler-Architektur:** HDX 3D Pro unterstützt die NVIDIA GRID K1- und K2-Karten für GPU-Passthrough und die gemeinsame GPU-Verwendung. Der NVIDIA GRID-vGPU ermöglicht mehreren VMs den gleichzeitigen direkten Zugriff auf einen physischen GPU und die Verwendung derselben NVIDIA-Grafiktreiber, die auf nicht-virtualisierten Betriebssystemen bereitgestellt werden.
- **Unterstützung für VMware vSphere und VMware ESX mit Virtual Direct Graphics Acceleration (vDGA):** Sie können HDX 3D Pro mit vDGA sowohl für Remotedesktopdienste- als auch für VDI-Arbeitslasten verwenden. Wenn HDX 3D Pro mit Virtual Shared Graphics Acceleration (vSGA) verwendet wird, wird nur ein Monitor unterstützt. Aufgrund der API-Abfangtechnologie von vSGA können bei der Verwendung mit großen 3D-Modellen Leistungsprobleme auftreten. Weitere Informationen zu Problemen finden Sie unter [VMware vSphere 5.1 - Citrix Known Issues](#).

HDX 3D Pro-Integration

HDX 3D Pro kann in eine bestehende XenApp- bzw. XenDesktop-Infrastruktur integriert werden. Sie können grafische Anwendungen als Teil von gehosteten Anwendungen oder Desktops auf Desktopbetriebssystemmaschinen bereitstellen.

Wie in der folgenden Abbildung dargestellt:

- Der Hostcomputer muss sich in derselben Active Directory-Domäne wie Ihr Delivery Controller befinden.
- Wenn sich ein Benutzer bei Citrix Receiver anmeldet und auf die virtuelle Anwendung oder den Desktop zugreift, authentifiziert der Controller den Benutzer und kontaktiert den VDA für HDX 3D Pro, um eine Verbindung zum Computer, der die grafische Anwendung hostet, zu vermitteln.
Der VDA für HDX 3D Pro komprimiert mit der entsprechenden Hardware auf dem Host die Ansicht des gesamten Desktops oder nur der grafischen Anwendung.
- Die Desktop- oder Anwendungsansichten und die dazugehörigen Interaktionen der Benutzer werden zwischen dem Hostcomputer und dem Benutzergerät über eine direkte HDX-Verbindung zwischen Citrix Receiver und VDA für HDX 3D Pro übertragen.



Installieren und Konfigurieren von HDX 3D Pro für Windows-Desktopbetriebssysteme

Voraussetzungen:

- Eine installierte und konfigurierte XenApp- oder XenDesktop-Infrastruktur
- Eine XenApp- oder XenDesktop-Site
- Informationen zur Verwendung von XenApp oder XenDesktop mit XenServer-vGPU finden Sie in
— *vGPU Release Notes*
und
— *Configuring XenServer to use NVIDIA GRID*
unter [Citrix Virtual GPU Solution](#).

Installieren über die grafische Oberfläche

Installieren Sie den Virtual Delivery Agent (VDA) für HDX 3D Pro, damit Benutzer eine Verbindung mit der physischen Maschine oder der XenServer-VM, die die grafische Anwendung hostet, herstellen können. Weisen Sie anschließend den virtuellen Desktop oder die virtuelle Anwendung einem Benutzer zu, indem Sie einen Maschinenkatalog und eine Bereitstellungsgruppe mit dem Computer erstellen, der die Grafikanwendung hostet.

1. Bereiten Sie die VMs oder physischen Maschinen vor, auf denen die Grafikanwendung ausgeführt wird: Installieren und richten Sie die Grafikanwendung und die anderen benötigten Anwendungen ein.
Weitere Informationen zum Zuweisen von GPUs zu XenServer-VMs finden Sie in [XenServer Virtual Machine User's Guide](#).
Hinweis: Standardmäßig wird jede virtuelle CPU, die einer XenServer-VM zugeordnet wird, einem Single-Core Socket zugewiesen. Dies bedeutet, dass Sie unter Betriebssystemen mit Sockelbeschränkungen nur eine begrenzte Anzahl der CPU-Kerne auf dem Hostserver verwenden können. XenServer Advanced, Enterprise und Platinum Edition enthalten ein Feature, mit dem Sie die Anzahl der Kerne pro virtuelle CPU in einer VM angeben können. Weitere Informationen finden Sie unter <http://support.citrix.com/article/CTX126524>.
2. Überprüfen Sie für Benutzerverbindungen mit physischen VDAs, an die physische Monitore angeschlossen sind, dass die Auflösung auf dem Benutzergerät der VDA-Auflösung entspricht oder geringer ist.
Wenn die vom Benutzergerät angeforderte Auflösung hoch ist, erkennt die ICA-Sitzung nicht die Auflösung des Benutzergeräts und verwendet die Mindestauflösung, die vom VDA-Monitor unterstützt wird.

3. Binden Sie den Hostcomputer in der Active Directory-Domäne ein, die Ihren Delivery-Controller enthält. Notieren Sie den Active Directory-Computerkontonamen des Hostcomputers, da Sie ihn beim Erstellen eines Maschinenkatalogs kennen müssen.
4. Legen Sie das Installationsmedium in das optische Laufwerk ein oder stellen Sie die ISO-Datei auf dem Hostcomputer bereit. Wenn Autorun nicht aktiviert ist, navigieren Sie zu AutoSelect.exe auf dem Installationsmedium und führen Sie das Programm aus.
5. Klicken Sie im Installationsassistenten auf Virtual Delivery Agent für Windows-Desktopbetriebssysteme oder Virtual Delivery Agent für Windows-Serverbetriebssysteme.
6. Wählen Sie auf der Seite "Umgebung" die Option Masterimage erstellen (wenn Sie Maschinenerstellungsdienste oder Provisioning Services zum Erstellen virtueller Desktops verwenden) oder Remote-PC-Zugriff aktivieren (wenn Sie möchten, dass Benutzer eine Verbindung zu einer vorhandenen Maschine herstellen).
7. Klicken Sie auf der Seite der Kernkomponenten auf Weiter.
8. Geben Sie auf der Seite der Delivery Controller die Controller der Site an, mit denen der VDA für HDX 3D Pro eine Verbindung herstellt. Geben Sie dazu die Standorte manuell ein oder wählen Sie die Controller aus Active Directory aus. Sie können die Controller auch durch Maschinenerstellungsdienste angeben lassen. Alternativ können Sie später auswählen, wenn Sie die Controller-Standorte zu einem späteren Zeitpunkt mit der Gruppenrichtlinie oder der erneuten Ausführung des Installers angeben möchten.

Wichtig: Stellen Sie sicher, dass Sie die Standorte aller Controller der Site angeben, da andernfalls einige Benutzerverbindungen abgelehnt werden könnten. Für den Lastausgleich verteilt VDA für HDX 3D Pro die Verbindungen automatisch gleichmäßig auf die Controller.
9. Wenn Sie auf dem Hostcomputer eine andere Firewall als die Windows-Firewall verwenden, aktivieren Sie die Ports 80, 1494, 2598 und 3389 manuell, damit XenApp bzw. XenDesktop richtig funktioniert. Öffnen Sie die Ports 16500 - 16509, um die Echtzeitübertragung für Audio zu aktivieren. Wenn die Windows-Firewall auf dem Hostcomputer ausgeführt wird, können Sie mit dem Installer die Ports automatisch öffnen lassen. Klicken Sie auf Next.
10. Klicken Sie auf der Seite Zusammenfassung auf Installieren.

Vor der Installation des VDA für HDX 3D Pro werden folgende Voraussetzungen installiert, wenn sie nicht bereits auf dem Hostcomputer vorhanden sind:

 - Microsoft .NET Framework 3.5 SP1
 - Visual C++ 2005, 2008 SP 1 und 2010 Redistributable-Pakete
11. Stellen Sie nach Abschluss der Installation sicher, dass das Kontrollkästchen Maschine neu starten (für den Abschluss der Installation erforderlich) aktiviert ist, und klicken Sie auf Schließen.
12. Nach dem Abschluss der Installation des VDA für HDX 3D Pro melden Sie sich bei dem Computer an, auf dem Studio ausgeführt wird, erstellen Sie einen Maschinenkatalog und fügen Sie den Computer, der die grafische Anwendung hostet, hinzu. Weitere Informationen zu diesem Schritt finden Sie unter [Erstellen eines neuen Maschinenkatalogs](#).

Verwenden Sie beim Erstellen eines Maschinenkatalogs für vGPU-fähige Desktops die folgenden Einstellungen:

 - Wählen Sie auf der Seite "Betriebssystem und Hardware" die Option Windows-Desktopbetriebssystem aus.
 - Klicken Sie auf der Seite "Maschinenverwaltung" auf VMs.
 - Wählen Sie auf der Seite "Desktopperfahrung" Benutzer sollen bei jeder Anmeldung mit dem gleichen (statischen) Desktop verbunden werden.
13. Damit die gehosteten Desktops auf Desktopbetriebssystemmaschinen den Benutzern zur Verfügung stehen, erstellen Sie eine Desktopbereitstellungsgruppe mit dem Maschinenkatalog, der den Hostcomputer enthält. Weitere Informationen zu diesem Schritt finden Sie unter [Erstellen einer neuen Bereitstellungsgruppe](#).

Installieren an der Befehlszeile

Sie können den VDA für HDX 3D Pro auch über eine Eingabeaufforderung installieren. Führen Sie zum Installieren von HDX 3D Pro XenDesktopVdaSetup.exe aus und geben Sie zusätzlich zu anderen evtl. benötigten Argumenten die

folgenden Optionen an.

- /ENABLE_HDX_3D_PRO

Weitere Informationen zu anderen Optionen finden Sie unter [Installieren an der Befehlszeile](#) oder indem Sie XenDesktopVdaSetup.exe/h ausführen.

Wenn Sie Delivery Agent für HDX 3D Pro über eine Active Directory-Gruppenrichtlinie bereitstellen, muss die Transformationsdatei entsprechende Werte für die Eigenschaften ENABLE_HDX_3D_PRO enthalten. Weitere Informationen zum Bereitstellen von VDA über eine Gruppenrichtlinie finden Sie unter [Installieren oder Entfernen von Virtual Delivery Agents in Active Directory mit Skripts](#).

Aktualisieren von HDX 3D Pro

Für das Upgrade von HDX 3D Pro müssen Sie die separate Komponente HDX 3D für professionelle Grafiken und den VDA deinstallieren, bevor Sie die aktuelle Version des VDA für HDX 3D Pro installieren. Ebenso müssen Sie zum Wechsel vom standardmäßigen VDA zu VDA für HDX 3D Pro den standardmäßigen VDA vor der Installation von VDA für HDX 3D Pro deinstallieren.

Installieren und Aktualisieren von NVIDIA-Treibern

Die NVIDIA GRID-API bietet direkten Zugriff auf den Framepuffer des Grafikprozessors und bietet die schnellste Framerate für eine gleichmäßige und interaktive Benutzererfahrung. Wenn Sie NVIDIA-Treiber vor einem VDA mit HDX 3D Pro installieren, ist NVIDIA GRID standardmäßig aktiviert.

Zum Aktivieren von NVIDIA GRID auf einer virtuellen Maschine müssen Sie den Microsoft Basic Display Adapter im Geräte-Manager deaktivieren. Führen Sie den folgenden Befehl aus und starten Sie dann den VDA neu:

`Montereyenable.exe –enable –noreset`

Wenn Sie NVIDIA-Treiber nach einem VDA mit HDX 3D Pro installieren, ist NVIDIA GRID deaktiviert. Aktivieren Sie NVIDIA GRID mit dem von NVIDIA bereitgestellten Tool Montereyenable.

Führen Sie den folgenden Befehl aus, um NVIDIA GRID zu aktivieren und starten Sie dann VDA neu:

`Montereyenable.exe –enable –noreset`

Führen Sie den folgenden Befehl aus, um NVIDIA GRID zu deaktivieren, und starten Sie dann den VDA neu:

`Montereyenable.exe –disable –noreset`

Optimierung der HDX 3D Pro-Benutzererfahrung

Implementieren Sie die folgenden Empfehlungen, um Ihren Benutzern eine optimale Benutzererfahrung zu bieten.

- Stellen Sie bei der Verwendung von HDX 3D Pro mit mehreren Monitoren sicher, dass der Hostcomputer mit mindestens so vielen Monitoren konfiguriert ist, wie an den Geräten der Benutzer angeschlossen sind. Die an den Hostcomputer angeschlossenen Monitore können physikalische oder virtuelle Monitore sein.
- Schließen Sie Monitore (physikalische oder virtuelle) nicht an Hostcomputer an, während Benutzer mit dem virtuellen Desktop oder der virtuellen Anwendung, die die grafische Anwendung bereitstellen, verbunden sind. Dies kann für die Dauer der Benutzersitzung zu Instabilität führen.
- Teilen Sie den Benutzern mit, dass das Ausführen von Änderungen (von ihnen oder einer Anwendung) an der Desktopauflösung, während eine grafische Anwendungssitzung ausgeführt wird, nicht unterstützt wird. Nach dem Beenden der Anwendungssitzung können Benutzer die Auflösung des Desktop Viewer-Fensters in "Citrix Receiver - Desktop Viewer-Einstellungen" ändern.
- Wenn mehrere Benutzer eine Verbindung mit beschränkter Bandbreite gemeinsam verwenden, z. B. in einer Zweigstelle,

empfiehlt Citrix die Verwendung der Richtlinie Bandbreitenlimit für Sitzung insgesamt, um die für die einzelnen Benutzer verfügbare Bandbreite zu beschränken. Damit wird sichergestellt, dass die verfügbare Bandbreite beim Anmelden und Abmelden der Benutzer keinen großen Schwankungen unterworfen ist. Da HDX 3D Pro automatische Anpassungen durchführt, um die gesamte Bandbreite auszuschöpfen, kann sich die stark variierende verfügbare Bandbreite während der Benutzersitzungen negativ auf die Leistung auswirken. Weitere Informationen finden Sie unter

[Richtlinieneinstellungen für die Bandbreite](#).

Wenn beispielsweise 20 Benutzer eine Verbindung mit 60 MBit/s gemeinsam verwenden, kann die Bandbreite, die den einzelnen Benutzern zur Verfügung steht, abhängig von der Anzahl der gleichzeitigen Benutzer zwischen 3 MBit/s und 60 MBit/s variieren. Um die Benutzererfahrung in diesem Szenario zu optimieren, legen Sie die Bandbreite fest, die zu Spitzenzeiten pro Benutzer erforderlich ist, und stellen Sie sicher, dass die Benutzer diesen Wert nicht überschreiten können.

- Citrix empfiehlt Benutzern einer 3D-Maus, die Priorität des virtuellen Kanals für die generische USB-Umleitung auf 0 zu erhöhen. Weitere Informationen dazu, wie Sie die Priorität virtueller Kanäle ändern können, finden Sie unter <http://support.citrix.com/article/CTX128190>.

GPU-Beschleunigung für Windows-Serverbetriebssystem

Jul 22, 2016

Mit HDX 3D Pro können grafikintensive Anwendungen, die in Windows-Serverbetriebssystemsitzen ausgeführt werden, auf der GPU des Servers wiedergegeben werden. Beim Verlagern der Wiedergabe von OpenGL, DirectX, Direct3D und Windows Presentation Foundation auf den Grafikprozessor des Servers wird die CPU des Servers nicht durch die Grafikwiedergabe verlangsamt. Zusätzlich kann der Server so mehr Grafiken verarbeiten, weil die Arbeitslast zwischen Prozessor und Grafikprozessor aufgeteilt wird.

Bei Verwendung von HDX 3D Pro können mehrere Benutzer eine Grafikkarte gemeinsam verwenden. Wenn HDX 3D Pro mit XenServer GPU-Passthrough verwendet wird, hostet ein Server mehrere Grafikkarten, d. h. eine pro virtuelle Maschine.

GPU-Freigabe für Remotedesktopdienste-Arbeitslasten

Die GPU-Freigabe ermöglicht die GPU-Hardwarewiedergabe von OpenGL- und DirectX-Anwendungen in Remotedesktopsitzen. Die GPU-Freigabe hat die folgenden Merkmale:

- Verwenden auf Bare-Metal- oder virtuellen Maschinen, um die Anwendungsskalierbarkeit und -leistung zu steigern.
- Mehrere gleichzeitige Sitzungen verwenden GPU-Ressourcen gemeinsam. (Die meisten Benutzer benötigen nicht die Wiedergabeleistung eines dedizierten GPU.)
- Erfordert keine besonderen Einstellungen.

Grafikkarten

Sie können mehrere GPUs auf einem Hypervisor installieren und VMs jeder dieser GPUs einzeln zuweisen.

- Installieren Sie eine Grafikkarte mit mehreren GPUs.
 - Oder installieren Sie mehrere Grafikkarten mit jeweils einem oder mehreren Grafikprozessoren.
- Eine Mischung heterogener Grafikkarten auf einem Server wird nicht empfohlen.

Hinweis: Virtuelle Maschinen benötigen einen direkten Passthrough-Zugriff auf einen GPU; dies ist mit Citrix XenServer oder VMware vSphere verfügbar. Wenn HDX 3D zusammen mit GPU-Passthrough verwendet wird, unterstützt jeder Grafikprozessor des Servers eine virtuelle Maschine mit mehreren Benutzern.

GPU Sharing hängt nicht von einer bestimmten Grafikkarte ab.

- Wählen Sie bei Ausführung auf einem Hypervisor eine Hardwareplattform und Grafikkarten aus, die mit der Implementierung von GPU-Passthrough des Hypervisors kompatibel sind. Die Liste der Hardware, die Zertifizierungstests mit XenServer GPU-Passthrough bestanden hat, finden Sie unter [GPU-Passthroughgeräte](#).
- Bei Ausführung auf Bare-Metal sollte eine Grafikkarte vom Betriebssystem aktiviert sein. Wenn mehrere GPUs auf der Hardware installiert sind, deaktivieren Sie mit dem Device Manager alle außer einem.

Skalierbarkeit

Die Skalierbarkeit mit der GPU-Freigabe hängt von diesen Faktoren ab:

- Ausgeführte Anwendungen
- Verbrauchter Videospeicher
- Verarbeitungsleistung der Grafikkarte

Beispiel: Skalierbarkeitswerte im Bereich von 8 bis 10 Benutzern wurden für NVIDIA Q6000- und M2070Q-Karten berichtet, die Anwendungen wie ESRI ArcGIS ausführen. Diese Karten haben 6 GB Videospeicher. Neuere NVIDIA GRID-Karten haben 8 GB Videospeicher und eine wesentlich höhere Verarbeitungsleistung (mehr CUDA-Kerne). Bei NVIDIA GRID K2-Karten wurde bei bis zu 20 Benutzern pro GRID K2-Karte eine gute Leistung beobachtet. Bei anderen Anwendungen kann die Skalierbarkeit wesentlich höher liegen; sie können 32 gleichzeitige Benutzer bei einem Profi-GPU erreichen.

Hinweis: Einige Anwendungen handhaben fehlenden Videospeicher besser als andere. Wenn die Hardware stark überlastet wird, kann der Grafikkartentreiber instabil werden oder abstürzen. Schränken Sie die Anzahl der gleichzeitigen Benutzer ein, um diese Probleme zu vermeiden.

Sie können die GPU-Beschleunigung mit einem Tool von Drittanbietern bestätigen, z. B. GPU-Z. GPU-Z finden Sie unter <http://www.techpowerup.com/gpuz/>.

Wiedergabe von DirectX, Direct3D und WPF

Die Wiedergabe von DirectX, Direct3D und WPF steht nur auf Servern zur Verfügung, die einen Grafikprozessor haben, der eine Anzeigetreiberschnittstelle der Version 9ex, 10 oder 11 unterstützt.

- Unter Windows Server 2008 R2 sind für DirectX und Direct3D keine Sondereinstellungen erforderlich, um einen einzelnen GPU zu verwenden.
- Unter Windows Server 2012 verwenden Remotedesktopdienste-Sitzungen auf dem RD-Sitzungshostserver als Standardadapter den Microsoft Basic Render-Treiber. Um den GPU in RDS-Sitzungen unter Windows Server 2012 zu verwenden, aktivieren Sie die Einstellung Use the hardware default graphics adapter for all Remote Desktop Services sessions in der Gruppenrichtlinie Lokale Computerrichtlinie > Computerkonfiguration > Administrative Vorlagen > Windows-Komponenten > Remotedesktopdienste > Remotedesktop-Sitzungshost > Remotesitzungsumgebung.
- Um die Wiedergabe von WPF-Anwendungen mit der GPU des Servers zu aktivieren, müssen Sie in der Registrierung des Servers, der die Windows-Serverbetriebssystem Sitzungen ausführt, die folgenden Einstellungen erstellen:
 - [HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CtxHook\ApplInit_Dlls\ Multiple Monitor Hook]
"EnableWPFHook"=dword:00000001
 - [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\CtxHook\ApplInit_Dlls\ Multiple Monitor Hook]
"EnableWPFHook"=dword:00000001

Experimentelle GPU-Beschleunigungsfunktion für CUDA- oder OpenCL-Anwendungen

Dieses Release stellt auch die Unterstützung für eine experimentelle GPU-Beschleunigung von CUDA- und OpenCL-Anwendung bereit, die in einer Benutzersitzung ausgeführt werden. Diese Unterstützung ist in der Standardeinstellung deaktiviert, Sie können sie jedoch für Test- und Evaluierungszwecke aktivieren.

Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab.

1. Aktivieren Sie die folgenden Registrierungseinstellungen, um die experimentellen CUDA-Beschleunigungsfeatures zu verwenden:
 - [HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CtxHook\ApplInit_Dlls\Graphics Helper] "CUDA"=dword:00000001
 - [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\CtxHook\ApplInit_Dlls\Graphics Helper]
"CUDA"=dword:00000001
2. Aktivieren Sie die folgenden Registrierungseinstellungen, um die experimentellen OpenCL-Beschleunigungsfeatures zu verwenden:
 - [HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CtxHook\ApplInit_Dlls\Graphics Helper] "OpenCL"=dword:00000001
 - [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\CtxHook\ApplInit_Dlls\Graphics Helper]

"OpenCL"=dword:00000001

OpenGL Software Accelerator

Sep 29, 2015

OpenGL Software Accelerator ist ein Softwarerasterizer für OpenGL-Anwendungen wie ArcGIS, Google Earth, Nehe, Maya, Blender, Voxler, CAD und CAM. In einigen Fällen bietet OpenGL Software Accelerator auch ohne die Verwendung einer Grafikkarte eine gute Benutzererfahrung mit OpenGL-Anwendungen.

Wichtig: OpenGL Software Accelerator wird ohne Modifikation bereitgestellt und muss für alle Anwendungen getestet werden. Er funktioniert möglicherweise mit einigen Anwendungen nicht, kann aber als Lösung dienen, wenn der Windows OpenGL-Rasterizer keine angemessene Leistung bietet. Wenn OpenGL Software Accelerator mit Ihren Anwendungen funktioniert, kann es Ihnen Kosten für GPU-Hardware sparen.

Unterstützte Plattformen:

- Windows 8, 32-Bit- und 64-Bit-Editionen
- Windows 7, 32-Bit- und 64-Bit-Editionen
- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2

Einsatzmöglichkeiten für OpenGL Software Accelerator

- Wenn die Leistung von OpenGL-Anwendungen in virtuellen Maschinen auf XenServer oder anderen Hypervisors unzureichend ist, verwenden Sie OpenGL Accelerator. Bei einigen Anwendungen werden mit OpenGL Accelerator durch den Einsatz von SSE4.1 und AVX bessere Ergebnisse erzielt als mit dem in Windows enthaltenen Microsoft OpenGL Rasterizer. OpenGL Accelerator unterstützt zudem Anwendungen, die OpenGL-Versionen bis 2.1 verwenden.
- Bei Anwendungen, die auf einer Arbeitsstation ausgeführt werden, sollte zunächst die Standardversion von OpenGL verwendet werden, die von der Arbeitsstation des Grafikadapters bereitgestellt wird. Wenn es sich um eine aktuelle Grafikkarte handelt, wird damit in den meisten Fällen die beste Leistung erzielt. Bei Grafikkarten älterer Versionen oder weniger leistungsstarken Grafikkarten sollte OpenGL Software Accelerator eingesetzt werden.
- 3D OpenGL-Anwendungen, die nicht angemessen bereitgestellt wurden und CPU-basierte Softwarerasterisierung verwenden, profitieren möglicherweise von der OpenGL GPU-Hardwarebeschleunigung. Dieses Feature kann auf Bare-Metal- oder virtuellen Maschinen verwendet werden. Weitere Informationen finden Sie unter [Gemeinsame GPU-Nutzung für OpenGL](#).

Optimieren von Audiofeatures

Sep 29, 2015

Um HDX-Audiofeatures zu optimieren, verwenden Sie die folgenden Citrix Richtlinienereinstellungen:

- Audioqualität
- Clientaudioumleitung
- Clientmikrofonumleitung
- Audio Plug & Play
- Bandbreitenlimit für die Audioumleitung
- Bandbreitenlimit für die Audioumleitung (Prozent)
- Audio über UDP - Real-time Transport
- Audio-UDP-Portbereich

Wichtig: Die meisten Audiofeatures werden über den ICA-Stream transportiert und werden auf dieselbe Weise gesichert wie anderer ICA-Datenverkehr. User Datagram Protocol (UDP)-Audio verwendet einen separaten, nicht gesicherten Transportmechanismus.

Auswahl der Audioqualität

Im Allgemeinen erfordert eine höhere Audioqualität mehr Bandbreite und führt zu einer höheren CPU-Auslastung, da mehr Audiodaten an die Benutzergeräte gesendet werden. Mit der Audiokomprimierung können Sie die Audioqualität und die Sitzungsleistung aufeinander abstimmen; verwenden Sie Citrix Richtlinienereinstellungen, um den Komprimierungsgrad für Audiodateien zu konfigurieren.

Standardmäßig ist die Audioqualität auf Hoch - High Definition-Audio eingestellt. Diese Einstellung bietet HiFi-Stereo-Audio, verbraucht aber mehr Bandbreite als die anderen Einstellungen.

Hinweis: Verwenden Sie diese Audioqualität nicht für nicht optimierte Chat- und Videochat-Anwendungen (z. B. Softphones), da es hierbei zu Verzögerungen im Audiopfad kommen kann, was bei Echtzeitkommunikation ungeeignet ist. Sie können auch die Erstellung separater Richtlinien für Benutzergruppen mit Einwählverbindungen und für Benutzer, die über ein LAN oder WAN eine Verbindung herstellen, in Erwägung ziehen. Bei Einwählverbindungen, bei denen die Bandbreite im Allgemeinen begrenzt ist, ist die Download-Geschwindigkeit den Benutzern gewöhnlich wichtiger als die Audioqualität. Erstellen Sie für solche Benutzer eine Richtlinie für DFÜ-Verbindungen, die für Audio einen hohen Komprimierungsgrad anwendet, sowie eine weitere Richtlinie für LAN- oder WAN-Verbindungen, die einen geringen Komprimierungsgrad anwendet.

Zum Ändern der Audioqualität fügen Sie die Einstellung Audioqualität einer Richtlinie hinzu und wählen Sie eine der folgenden Audioqualitätsstufen:

- Mittel - optimiert für Sprache: Verwenden Sie diese Einstellung, um VoIP-Anwendungen bereitzustellen, bzw. zum Bereitstellen von Medienanwendungen bei schwierigen Netzwerkverbindungen mit sehr niedriger Leistung (weniger als 512 KBit/s) oder bei erheblicher Überlastung und Paketverlust. Die an das Benutzergerät gesendeten Audiodaten werden bis auf 64 Kbit/s komprimiert; diese Komprimierung führt zu einer moderaten Verringerung der Tonqualität auf dem Benutzergerät mit niedriger Latenz und geringem Bandbreitenverbrauch. Wenn die Einstellung Mittel für die Audioqualität eine unbefriedigende VoIP-Qualität liefert, stellen Sie sicher, dass die Richtlinie Audio über UDP Real-time Transport auf Zugelassen eingestellt ist.
Hinweis: UDP/RTP erfordert die Einstellung Mittel für die Audioqualität.
- Gering - für langsame Verbindungen: Die an das Benutzergerät gesendeten Audiodaten werden bis auf 16 KBit/s komprimiert; diese Komprimierung führt zu einer erheblichen Verringerung der Tonqualität, liefert aber eine angemessene Leistung auch bei Verbindungen mit geringer Bandbreite.

Wichtig: Denken Sie daran, die Clientaudioeinstellungen auf dem Benutzergerät für Audio zu aktivieren (wie weiter unten beschrieben).

Umleiten des Audioempfangs

Sie können den Audioempfang von einer Anwendung auf dem Server über Lautsprecher oder andere Soundgeräte (z. B. Kopfhörer) auf dem Benutzergerät zulassen. Standardmäßig ist die Clientaudioumleitung zugelassen.

Wichtig: Um dieses Feature zu verwenden, müssen Sie die Clientaudioeinstellungen auf dem Benutzergerät für Audio aktivieren (wie weiter unten beschrieben). Auf Windows-Serverbetriebssystemmaschinen müssen Sie auch sicherstellen, dass für Audio Plug & Play die Unterstützung mehrerer Audiogeräte aktiviert ist.

Die Clientaudiozuordnung kann dazu führen, dass eine hohe Last auf dem Server und dem Netzwerk entsteht; um dieses Feature zu deaktivieren, fügen Sie die Einstellung Clientaudioumleitung einer Richtlinie hinzu und stellen Sie den Wert auf Nicht zugelassen ein.

Wichtig: Wenn die Clientaudioumleitung auf Nicht zugelassen eingestellt ist, sind alle HDX-Audiofunktionen auch deaktiviert.

Aktivieren der Mikrofone des Benutzergeräts

Sie können auch das benutzerseitige Aufzeichnen von Audio mit Eingabegeräten auf dem Benutzergerät, z. B. einem Mikrofon, ermöglichen. Standardmäßig ist die Clientmikrofonumleitung auf Zugelassen eingestellt. Wenn Audio in der Clientsoftware deaktiviert ist, oder wenn Clientaudioumleitung auf Nicht zugelassen eingestellt ist, hat diese Einstellung keine Auswirkung.

Für die Audioaufzeichnung muss das Benutzergerät entweder über ein integriertes Mikrofon oder ein Gerät verfügen, das in die Mikrofonbuchse oder den USB-Anschluss gesteckt werden kann. Aus Sicherheitsgründen werden Benutzer darauf hingewiesen, wenn Server, die keine vertrauenswürdige Beziehung zu den Benutzergeräten haben, auf Mikrofone zugreifen und können dann den Zugriff vor Verwenden des Mikrofons akzeptieren oder ablehnen. Benutzer können die diesbezügliche Warnung in Citrix Receiver deaktivieren.

Wichtig: Um dieses Feature zu verwenden, müssen Sie die Clientaudioeinstellungen auf dem Benutzergerät für Audio aktivieren (wie weiter unten beschrieben). Auf Windows-Serverbetriebssystemmaschinen müssen Sie auch sicherstellen, dass für Audio Plug & Play die Unterstützung mehrerer Audiogeräte aktiviert ist.

Zum Deaktivieren dieses Features fügen Sie die Einstellung Clientmikrofonumleitung einer Richtlinie hinzu und stellen Sie den Wert auf Nicht zugelassen ein.

Zulassen mehrerer Audiogeräte

Die Richtlinie Audio Plug & Play steuert, ob mehrere Audiogeräte zum Aufzeichnen und Wiedergeben verwendet werden dürfen. Diese Einstellung ist standardmäßig aktiviert; um dieses Feature zu deaktivieren, fügen Sie die Einstellung Audio Plug & Play einer Richtlinie hinzu und stellen Sie den Wert auf Deaktiviert ein.

Hinweis: Diese Einstellung gilt nur für Windows-Serverbetriebssystemmaschinen.

Festlegen von Bandbreitenlimits für die Audioumleitung

Die Richtlinieneinstellung Bandbreitenlimit für die Audioumleitung gibt die maximale Bandbreite (in Kilobits pro Sekunde) für die Wiedergabe und Aufzeichnung von Audio in einer Sitzung an. Die Einstellung Bandbreitenlimit für die Audioumleitung (Prozent) gibt die maximale Bandbreite für die Umleitung als Prozentsatz der insgesamt verfügbaren Bandbreite an. Standardmäßig ist Null (Maximum) für beide Einstellungen angegeben.

Zum Konfigurieren von Bandbreitenlimit für die Audioumleitung oder Bandbreitenlimit für die Audioumleitung (Prozent) oder beides, fügen Sie die Einstellung einer Richtlinie hinzu und geben Sie eine Zahl in das entsprechende Feld Wert ein. Wenn beide Einstellungen konfiguriert sind, wird die Einstellung mit dem niedrigsten Bandbreitenlimit verwendet.

Wichtig: Um dieses Feature zu verwenden, müssen Sie die Clientaudioeinstellungen auf dem Benutzergerät für Audio aktivieren (wie weiter unten beschrieben).

Senden und Empfangen von Audio mit UDP

Standardmäßig ist Audio über UDP - Real-time Transport auf Zugelassen eingestellt, sodass ein UDP-Port auf dem Server für alle Verbindungen geöffnet wird, die für Echtzeitübertragung von Audio über UDP konfiguriert wurden. Citrix empfiehlt, dass Sie UDP/RTP für Audio konfigurieren, um auch bei Netzwerküberlastung oder Paketverlust die beste Benutzererfahrung sicherzustellen.

Wichtig: Mit UDP übertragene Audiodaten sind unverschlüsselt.

Standardmäßig verwendet UDP-Audio auf XenDesktop zwei aufeinander folgende Ports zwischen Ports 16500 und 16509, um durch die Windows-Firewall zu gelangen; die Standardports sind 16500,16501. Um andere Ports zu verwenden, fügen Sie die Einstellung Audio-UDP-Portbereich einer Richtlinie hinzu und geben Sie die Portnummer oder den Bereich (im Format niedrigste Portnummer, höchste Portnummer) in das Feld Wert ein. Mit dieser Einstellung geben Sie den Bereich der Portnummern an, die von Virtual Desktop Agent zum Austausch von Audiopaketen mit dem Benutzergerät verwendet werden. Der VDA versucht, jedes UDP-Portpaar für den Austausch von Daten mit dem Benutzergerät zu verwenden. Dabei wird mit dem Port, der die niedrigste Nummer hat, begonnen und die Zahl für jeden folgenden Versuch um 2 erhöht. Alle angegebenen Ports übernehmen eingehende und ausgehende Datenübertragungen.

Wichtig: Um Audio über UDP zu verwenden, müssen Sie die Clientaudioeinstellungen auf dem Benutzergerät für Audio aktivieren (wie weiter unten beschrieben).

Konfigurieren der Audioeinstellungsrichtlinien für Benutzergeräte

1. Erweitern Sie Administrative Vorlagen > Klassische administrative Vorlage (ADM) > Citrix Komponenten > Citrix Receiver > Benutzerauthentifizierung.
2. Wählen Sie für Clientaudioeinstellungen die Option Nicht konfiguriert, Aktiviert oder Deaktiviert.
3. Wenn Sie Aktiviert eingestellt haben, wählen Sie eine Tonqualität. Verwenden Sie für UDP-Audio nur die Standardeinstellung Mittel.
4. Wählen Sie nur für UDP-Audio die Einstellung Echtzeitübertragung aktivieren und legen Sie den Bereich der eingehenden Ports so fest, dass der Durchgang durch die lokale Windows Firewall gewährleistet ist.

Vermeiden von Echo in Multimediakonferenzen

Wenn Benutzer an Audio- oder Videokonferenzen teilnehmen, hören sie ggf. ein Echo im Audio. Echos treten normalerweise auf, wenn der Abstand zwischen Lautsprechern und Mikrofonen nicht groß genug ist. Aus diesem Grund empfiehlt Citrix, dass Sie für Audio- und Videokonferenzen Kopfhörer verwenden.

HDX bietet eine Option für die Echounterdrückung, die in der Standardeinstellung aktiviert ist und das Auftreten von Echo in einer Konferenz minimiert. Die Qualität der Echounterdrückung hängt stark vom Abstand zwischen den Lautsprechern und dem Mikrofon ab. Der Abstand zwischen den Geräten darf nicht zu groß aber auch nicht zu klein sein.

Deaktivieren der Echounterdrückung

1. Navigieren Sie zu dem entsprechenden Registrierungseintrag.
Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab.
 - **32-Bit-Computer:** Öffnen Sie die Registrierung auf dem Benutzergerät und navigieren Sie zu

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\ICA
Client\Engine\Configuration\Advanced\Modules\ClientAudio\EchoCancellation.

- **64-Bit-Computer:** Öffnen Sie die Registrierung auf dem Benutzergerät und navigieren Sie zu
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\ICA
Client\Engine\Configuration\Advanced\Modules\ClientAudio\EchoCancellation.

2. Ändern Sie den Wert im Feld Value data zu FALSE.

Zuweisen von Prioritäten für den Datenverkehr im Netzwerk

Sep 29, 2015

Prioritäten für den Netzwerkdatenverkehr über mehrere Verbindungen für eine Sitzung werden zugewiesen, indem Diensteigenschaft (QoS)-fähige Router verwendet werden. Vier TCP/IP-Streams (real-time, interactive, background und bulk) und ein UDP/RTP-Stream (für Voice) sind zum Ausführen von ICA-Datenverkehr zwischen dem Benutzergerät und dem Server verfügbar. Jeder virtuelle Kanal ist mit einer bestimmten Priorität verknüpft und wird von der entsprechenden TCP-Verbindung transportiert. Sie können die Kanäle basierend auf der Portnummer, die für die Verbindung verwendet wird, unabhängig voneinander festlegen. Es gibt die folgenden vier Prioritäten:

- Sehr hoch: Für Echtzeitvorgänge, z. B. Webkonferenzen.
- Hoch: Für interaktive Elemente, z. B. Bildschirm, Tastatur und Maus.
- Mittel: Für Massenvorgänge, z. B. Clientlaufwerkzuordnung.
- Niedrig: Für Hintergrundaufgaben, z. B. Drucken.

XenApp und XenDesktop unterstützen Mehrkanal-Streamingverbindungen für Virtual Delivery Agents (VDAs) in Windows 8- und Windows 7-Umgebungen. Arbeiten Sie mit dem Netzwerkadministrator Ihres Unternehmens zusammen, um sicherzustellen, dass die in der Einstellung Multiport-Richtlinie konfigurierten Common Gateway Protocol (CGP)-Ports auf den Netzwerkroutern richtig zugewiesen sind.

Secure Sockets Layer (SSL)-Verbindungen werden nur unterstützt, wenn die Verbindungen durch ein NetScaler Gateway passieren, das Multistream unterstützt. Bei internen Unternehmensnetzwerken werden Multistreamverbindungen mit SSL nicht unterstützt.

Quality of Service wird nur unterstützt, wenn mehrere Sitzungszuverlässigkeitsports, oder CGP-Ports, konfiguriert sind.

Achtung: Verwenden Sie Transportsicherheit, wenn Sie dieses Feature einsetzen. Citrix empfiehlt die Verwendung von Internetprotokollsicherheit (IPsec) oder Secure Sockets Layer (SSL).

Einstellen der Audioqualität für Multistreamverbindungen

Fügen Sie die folgenden Einstellungen in einer Richtlinie hinzu:

1. Multiportrichtlinie: Diese Einstellung legt Ports für den ICA-Verkehr über mehrere Verbindungen fest und definiert die Netzwerkpriorität.
 1. Wählen Sie eine Priorität aus der Liste CGP-Standardportpriorität aus. Standardmäßig hat der primäre Port (2598) eine hohe Priorität.
 2. Geben Sie zusätzliche CGP-Ports in den Feldern CGP-Port1, CGP-Port2 und CGP-Port3 wie gewünscht ein und legen Sie die Priorität fest; jeder Port muss eine eindeutige Priorität haben.Wichtig: Firewalls auf VDAs müssen explizit so konfiguriert werden, dass zusätzlicher TCP-Datenverkehr zulässig ist.
2. Multistream: Diese Einstellung ist standardmäßig deaktiviert, da sie nicht erforderlich ist, wenn ein Branch Repeater für Multistream verwendet wird. Wählen Sie Aktiviert:
 - Erzielt gewünschte QoS bei Verwendung von Branch Repeatern oder Routern von Drittanbietern.
 - Aktiviert Multistream für bestimmte Benutzer.

Wichtig: Damit die Richtlinien wirksam werden, müssen sich Benutzer abmelden und dann am Netzwerk anmelden.

Konfigurieren des Dateizugriffs auf zugeordnete Clientlaufwerke

Sep 29, 2015

Sie können steuern, ob Benutzer Dateien von ihren virtuellen Umgebungen auf ihre Benutzergeräte kopieren können. Standardmäßig ist in der Sitzung Lese-/Schreibzugriff auf Dateien und Ordner auf zugeordneten Clientlaufwerken möglich.

Um zu verhindern, dass Benutzer Dateien und Ordner auf zugeordneten Clientlaufwerken hinzufügen oder ändern, aktivieren Sie die Richtlinieneinstellung Schreibgeschützter Zugriff auf Clientlaufwerke.

Wichtig: Wenn Sie diese Einstellung einer Richtlinie hinzufügen, müssen Sie auch für die Einstellung Clientlaufwerkumleitung die Option Zugelassen wählen.

Citrix Connector 7.5 für Configuration Manager

Sep 29, 2015

Citrix Connector 7.5 stellt eine Brücke zwischen Microsoft System Center Configuration Manager und XenApp oder XenDesktop bereit, sodass Sie die Verwendung von Configuration Manager auf Ihre Citrix Umgebungen ausweiten können. Connector 7.5 ist mit XenApp (7.6/7.5) und XenDesktop (7.6/7.5/7.1) kompatibel.

Mit Citrix Connector können Sie alltägliche Vorgänge übergreifend über die mit Configuration Manager verwalteten physischen Umgebungen und die mit XenApp oder XenDesktop verwalteten virtuellen Umgebungen vereinheitlichen. Der Connector bietet Folgendes:

- Einzelner Speicherort zum Definieren und Verwalten des Benutzerzugriffs auf alle Anwendungen
- Erweiterung der Bereitstellungsfunktionen von Configuration Manager, damit jede Anwendung jedem Benutzer auf praktisch jedem Gerät bereitgestellt werden kann
- Mehrwert für XenApp bzw. XenDesktop durch Verwendung von Configuration Manager-Richtlinien zur Automatisierung der Installation von Anwendungs- und Softwareupdates
- Durchgehend automatische und konsistente Softwareinstallation für eine oder mehrere Citrix Bereitstellungssites

Die Dokumentation zu Citrix Connector richtet sich an Administratoren, die mit Microsoft System Center Configuration Manager 2012 und XenApp bzw. XenDesktop 7.6/7.5 vertraut sind.

Neben der in diesem Abschnitt von eDocs veröffentlichten Dokumentation besuchen Sie auch das [Blog](#) und das [Forum](#) zu Citrix Connector.

[Info über Citrix Connector 7.5](#)

[Erstellen von Anwendungen](#)

[Systemanforderungen](#)

[Bereitstellen von Anwendungen](#)

[Planen der Bereitstellung](#)

[Veröffentlichen von Anwendungen](#)

[Installation und Einrichtung](#)

[Überwachen und Problembehandlung](#)

Info zu diesem Release

Sep 29, 2015

Mit dem Connector können Sie Configuration Manager für Folgendes verwenden:

- Synchronisieren von XenApp- oder XenDesktop-Maschinenkatalogen und -Bereitstellungsgruppen in Configuration Manager
- Bereitstellen von Software für XenApp- und XenDesktop-Maschinenkataloge
- Nutzung von MSI-, App-V- und Skript-Anwendungen, die bereits in Configuration Manager definiert sind
- Bereitstellen von Anwendungen für Maschinenkataloge, die über Maschinenerstellungsdienste (MCS) oder Provisioning Services verwaltet werden
- Verfolgen des Status von Anwendungsbereitstellungen
- Veröffentlichen von Anwendungen in Receiver auf jedem von XenApp oder XenDesktop unterstützten Benutzergerät
- Bereitstellen über Citrix gehosteter Anwendungen für den Configuration Manager-Anwendungskatalog oder das Softwarecenter auf mit Configuration Manager verwalteten Geräten
- Paralleles Arbeiten mit dem XenApp 6.5-Connector zum Erstellen von Zielanwendungen, Veröffentlichungselementen und zum Planen von Wartungsfenstern für alle Serversammlungen in einer heterogenen Umgebung über eine einzelne Configuration Manager-Konsole, auf der Citrix Connector 7.5 installiert ist

Dieses Video gibt Ihnen eine allgemeine Übersicht über die potenziellen Vorzüge von Citrix Connector für Ihre Organisation.

Bekannte Probleme

- Das Veröffentlichen von Anwendungen in administrativen Ordnern wird vom Connector in XenApp und XenDesktop 7.6 nicht unterstützt. Davon abgesehen ist Connector 7.5 vollständig mit XenApp und XenDesktop 7.6 kompatibel. Wenn der Connector eine Anwendung für eine Citrix Bereitstellungssite veröffentlicht und diese Anwendung dann mit Citrix Studio in einen anderen administrativen Ordner verschoben wird, kann der Connector den gewünschter Zustand der Veröffentlichung nicht unterhalten und Änderungen, die über Configuration Manager an der Veröffentlichung vorgenommen werden, schlagen fehl. Beim Connector-Veröffentlichungstask tritt eine Ausnahme nur für diese Veröffentlichung auf. Sie vermeiden dieses Problem, indem Sie zum Verschieben einer vom Connector veröffentlichten Anwendung nicht Studio verwenden. [#500630]
- Der Connector-Konfigurationsassistent wird möglicherweise nicht abgeschlossen, wenn die Configuration Manager-Site mehrere SMS-Anbieter enthält und nicht alle verfügbar sind. Reparieren oder entfernen Sie als Lösungsansatz die nicht verfügbaren SMS-Anbieter. [#316]

- Die Configuration Manager-Konsole kann keine Bereitstellungstechnologie laden, die nicht bei der Konsole registriert ist. Daher stürzt die Konsole bei den folgenden Vorgängen ab, wenn die Konsolenerweiterungen für den Connector 7.5 und den XenApp 6.5-Connector in derselben Bereitstellung verwendet werden:
 - Wenn Sie den Befehl zum Erstellen von Bereitstellungstypen für eine Anwendung verwenden, deren Bereitstellungstyp mit der anderen Connector-Konsolenerweiterung erstellt wird [#56]
 - Wenn Sie einen Bereitstellungstyp auswählen, der mit der anderen Connector-Konsolenerweiterung erstellt wird [#184]

Zur Problemlösung installieren Sie die fehlende Konsolenerweiterung.

- Wenn Sie die Connector-Konsolenerweiterung auf einer Maschine verwenden, auf der der Connector-Dienst nicht installiert ist, und das Dialogfeld "Eigenschaften" für einen Eintrag in der Liste der Citrix Application Publications öffnen, tritt ein Fehler auf. Problemumgehung: [#586]
 1. Wechseln Sie zu Anwendungsverwaltung > Anwendungen, wählen Sie eine Anwendung mit dem Bereitstellungstyp Citrix XenApp 7.5 und XenDesktop 7.5 aus und wählen Sie dann die Registerkarte Bereitstellungstypen. Das Feld "Technologie" ist für den Citrix Bereitstellungstyp leer.
 2. Doppelklicken Sie auf den Citrix Bereitstellungstypen-Eintrag. Nach etwa 10 Sekunden werden die Eigenschaften von Configuration Manager geladen. Sie können dann das Dialogfeld "Eigenschaften" für Veröffentlichungen öffnen.
- Der Orchestrierungstask wird nicht abgeschlossen und das Protokoll Citrix.ConfigMgr.OrchestrationTask enthält den Eintrag "System.Management.ManagementException: Quota violation". Um dieses Problem zu umgehen, vervierfachen Sie die Speicherlimits. Anweisungen hierzu finden Sie unter [WMI Error: 0x8004106C Description: Quota violation, while running WMI queries](#) [#705]
- Die Anweisung "Wählen Sie diese Option, wenn Sie den Citrix Connector für System Center Configuration Manager verwenden" auf der Seite "Rolloutstrategie" des Assistenten für Maschinenupdates trifft nicht auf alle Fälle zu. Wenn Sie MCS für Serverbetriebssystemmaschinenkataloge verwenden, orchestrieren Sie das Update von Imageklonen mit Studio.
- Bei einem mit der Befehlszeilenoption "/servervdi" installierten VDA ist die Eigenschaft "IsMasterImage" auf "false" festgelegt, außer es wird zusätzlich die Option "/masterimage" angegeben. Aus diesem Grund wird der zugehörige Katalog vom Connector nicht in der Liste "Designate Update Device" aufgeführt. Zur Problemumgehung installieren Sie den VDA erneut und verwenden Sie dabei die Befehlszeilenoption "/masterimage". [#489771]
- Eine Anwendung bleibt weiterhin veröffentlicht, nachdem Sie den Studio-Befehl **Rollback für Maschinenupdate** zur Wiederherstellung der vorherigen Version eines Masterimages verwendet haben. Zur Problemumgehung deaktivieren Sie die Anwendung bis die Maschinen kompatibel sind, oder entfernen Sie die Veröffentlichung aus Configuration Manager, wodurch sie aus Studio entfernt wird. Durch jede dieser Aktion wird die Anwendung für alle Benutzer betriebsunfähig. [#635]
- Wenn an den Eigenschaften einer Citrix Veröffentlichung von zwei Configuration Manager-Konsolen aus gleichzeitig Änderungen vorgenommen werden, wird nur ein Satz von Änderungen gespeichert. [#314]
- Bei einer Failoverkonfiguration mit Delivery Controller (Connector 1 verweist z. B. auf Delivery Controller A und Connector 2 auf Delivery Controller B) erfolgt kein Failover von Connector 1 auf Delivery Controller B, wenn Delivery Controller A heruntergefahren wird. Zur Problemumgehung halten Sie den Connector 1-Dienst manuell an, sodass ein Failover der Vorgänge zu Connector 2 und Delivery Controller B erfolgt. [#711]
- Der Connector erkennt eine Masterimage-VM nicht, auf der Windows XP oder Windows Vista ausgeführt wird. Ursache ist, dass der Connector eine Erkennung eines Masterimages durch XenApp oder den XenDesktop-VDA 7.1, 7.5 oder 7.6 erfordert. Diese VDA-Versionen unterstützen Windows XP und Windows Vista nicht. [#452]
- Nach der Deinstallation der Connector-Komponente "Citrix Gruppenrichtlinienverwaltung" vom Delivery Controller stürzt Studio ab, wenn Sie auf den Knoten "Richtlinien" in Studio klicken. Zur Problemumgehung installieren Sie das Citrix Richtlinien-Plug-In auf dem XenApp- bzw. XenDesktop-Installationsmedium neu. [#661]

Grundlegende Konzepte

In diesem Abschnitt finden Sie eine Übersicht über die Konzepte der XenApp- bzw. XenDesktop-Infrastruktur. Diese Informationen sind grundlegend dafür, die Funktionsweise von Citrix Connector und seine Unterscheidungsmerkmale im Vergleich zum [XenApp 6.5-Connector](#) kennenzulernen.

In unterstützten Versionen von XenApp und XenDesktop werden Ressourcen über eine Kombination aus Maschinenkatalogen und Bereitstellungsgruppen organisiert. In XenApp 6.5 werden Ordner und Workergruppen zum Organisieren von Anwendungen und Servern verwendet.

Maschinenkataloge

Maschinenkataloge sind Sammlungen von Maschinen mit derselben Konfiguration (z. B. Betriebssystem, Anwendungen und Desktopmodus).

- Kataloge für Desktopbetriebssystemmaschinen werden verwendet, um den Benutzern Anwendungen von Windows-Desktopbetriebssystemen oder generische oder personalisierte Desktops bereitzustellen.
- Mit Katalogen für Serverbetriebssystemmaschinen werden Benutzern Anwendungen oder Desktops zur Verfügung gestellt.
- Remote-PC-Zugriff-Maschinenkataloge werden verwendet, um Benutzern den Remotezugriff auf ihre physischen Bürodeshktops zu ermöglichen.

Der Connector synchronisiert zwar Remote-PC-Zugriff-Maschinenkataloge, er orchestriert jedoch keine Installation von Anwendungen oder Updates für sie.

Hinweis: Das Wake-On-LAN-Feature von Remote-PC-Zugriff erfordert Configuration Manager. Informationen finden Sie unter [Microsoft System Center Configuration Manager und Remote-PC-Zugriff-Wake-On-LAN](#).

Der Connector synchronisiert jeden Maschinenkatalog mit Configuration Manager als Gerätesammlung, sodass Sie Anwendungen dort bereitstellen können.

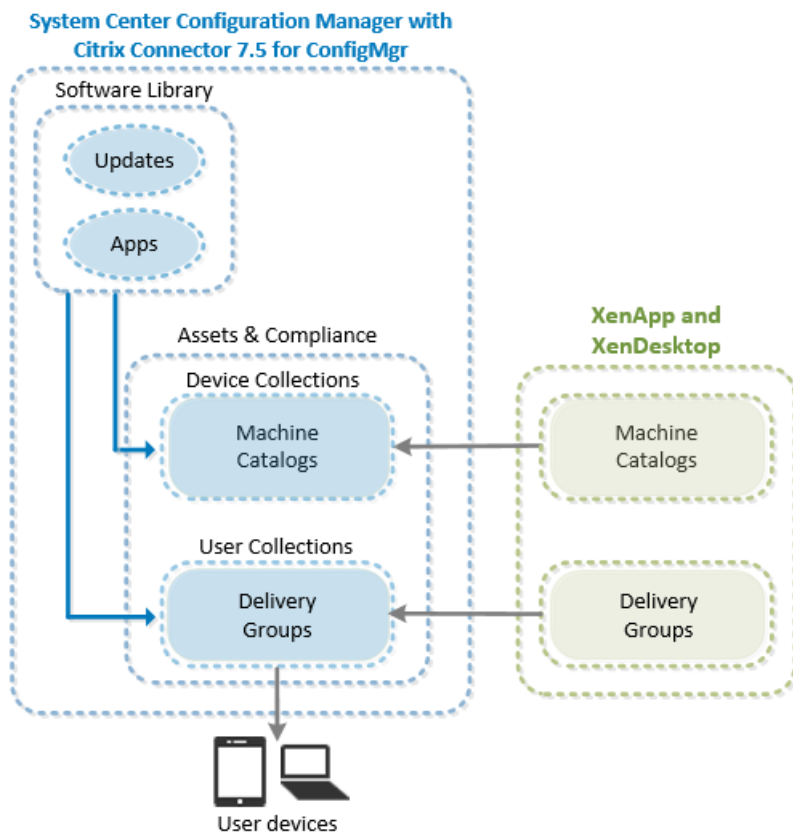
Die Maschinen in einem Maschinenkatalog sind die XenApp- bzw. XenDesktop-Worker (oder auch "Sitzungsmaschinen"). Für Provisioning Services und Maschinenerstellungsdienste (MCS) enthalten die Maschinenkataloge die VM für die Masterimages und die Maschinenklone.

Bereitstellungsgruppen

Bereitstellungsgruppen dienen zur Definition der Anwendungen und virtuellen Desktops, auf die eine Benutzergruppe zugreifen kann. Über Bereitstellungsgruppen wird auch festgelegt, welche Maschinenkataloge Anwendungen und Desktops für Benutzer bereitstellen. Die Beziehung zwischen Maschinenkatalogtypen und Bereitstellungsgruppen wird in der Dokumentation von [XenApp und XenDesktop](#) erläutert.

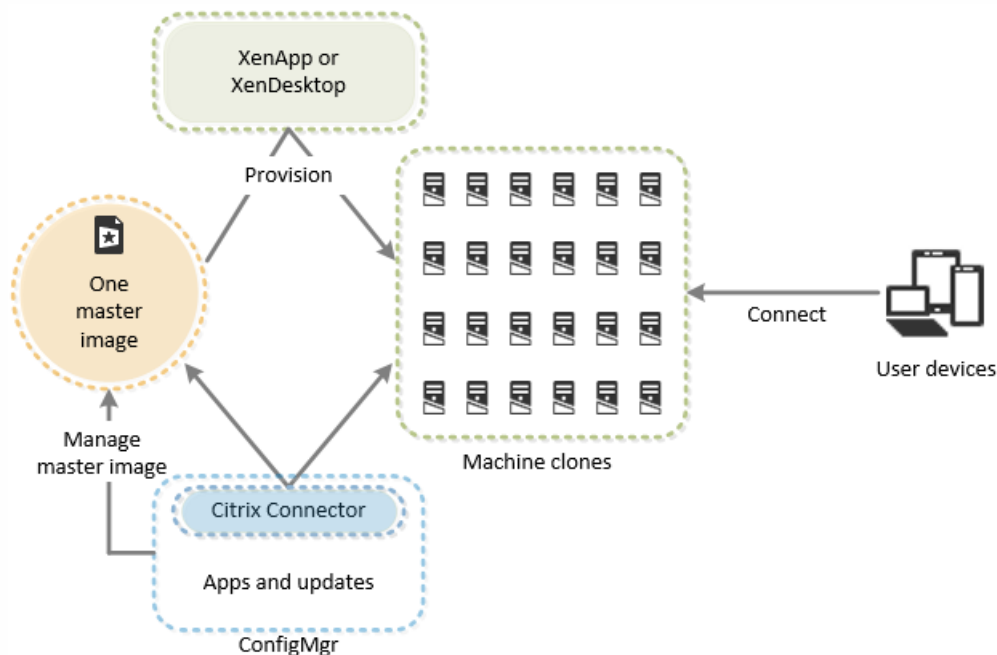
Der Connector synchronisiert jede Bereitstellungsgruppe mit Configuration Manager als Benutzersammlung.

Die folgende Abbildung zeigt die Beziehung zu Configuration Manager für die von XenApp bzw. XenDesktop verwalteten Maschinenkataloge und Bereitstellungsgruppen bei Verwendung des Connectors.



Verwaltung von Masterimages

XenApp und XenDesktop bieten eine zentrale Imageverwaltung, mit der Sie ein einzelnes Masterimage verwalten und aus diesem zahlreiche Sitzungsmaschinen ("Maschinenklone") bereitstellen können.



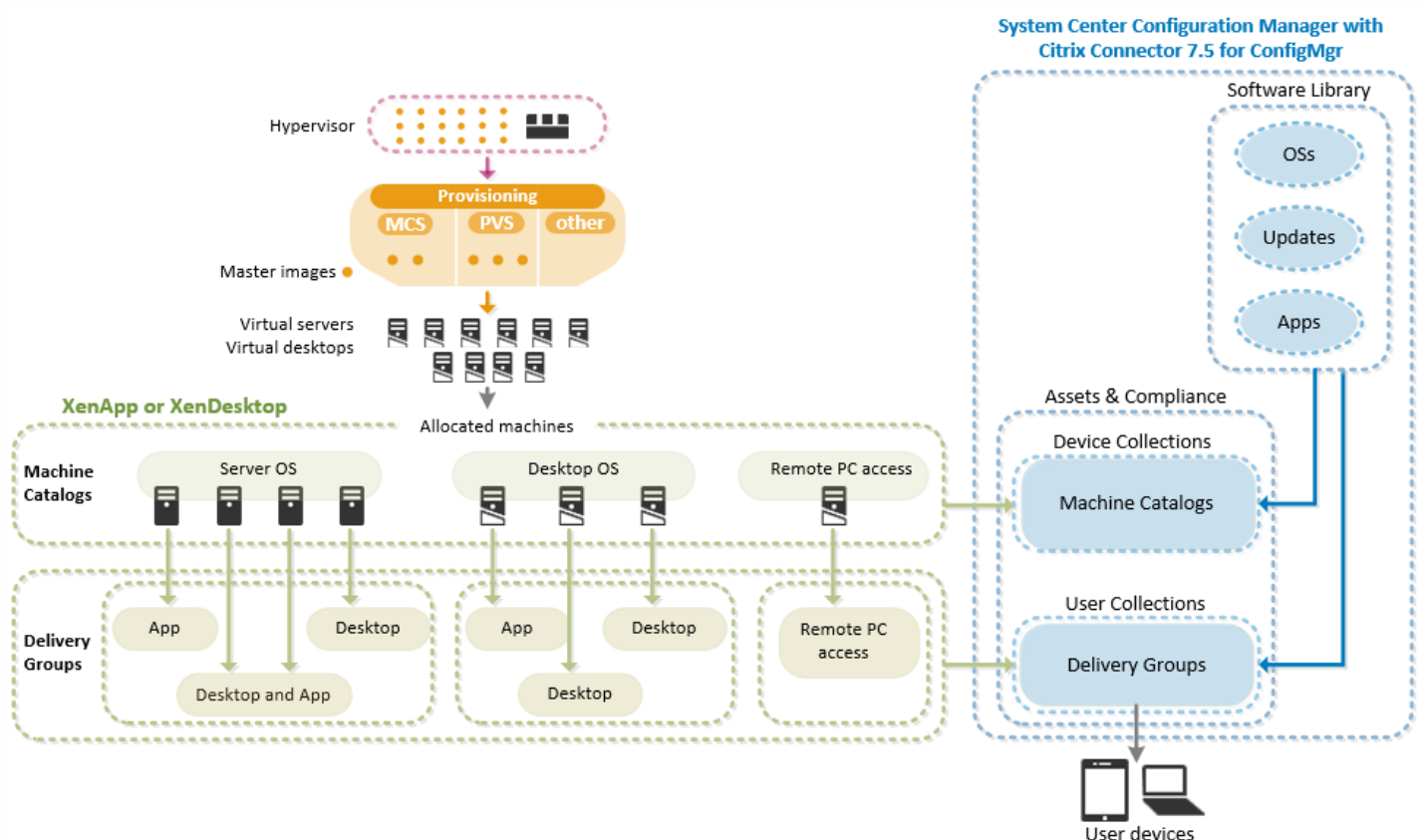
Ein Masterimage ist eine virtuelle Festplatte mit dem Betriebssystem. Ein Masterimage wird von einer Provisioningtechnik zur Erstellung von Maschinen verwendet, die Endbenutzern Anwendungen und Desktops zur Verfügung stellen. Abhängig von

der verwendeten Provisioningtechnologie, kann mit dem Masterimage auch eine Maschine erstellt werden, die Anwendungen und Desktops hostet.

Die Provisioningmethode wird vom Citrix Administrator beim Erstellen eines Maschinenkatalogs in XenApp bzw. XenDesktop gewählt. Der Connector unterstützt alle Methoden.

- **Maschinenerstellungsdienste:** Verwalten die virtuellen Maschinen mit einem Masterimage in der Umgebung, sodass Sie Zielgeräte über ein Masterimage verwalten und aktualisieren können.
- **Provisioning Services:** Ermöglicht ein Provisioning und erneutes Provisioning von Computern in Echtzeit von einem freigegebenen Datenträgerimage. Die Desktops und Anwendungen werden von einer Provisioning Services-vDisk bereitgestellt, deren Image von einem Masterzielgerät erstellt wurde. So können Sie die Verarbeitungsleistung der physischen Hardware oder der virtuellen Maschinen nutzen. Der Connector unterstützt mehrere Master-VMs, die für verschiedene Bereitstellungsgruppen jeweils andere Anwendungen bereitstellen dürfen.
- **Manuelle Bereitstellung:** Dient zur Verwaltung und Bereitstellung von Desktops und Anwendungen, die Sie bereits auf VMs im Rechenzentrum migriert haben. Sie können die Zielgeräte (Sitzungsmaschinen) einzeln oder zusammen mit Configuration Manager verwalten. Werden einem Maschinenkatalog Maschinenklone hinzugefügt, stellt Configuration Manager ihnen automatisch Anwendungen bereit. Daher müssen sich Citrix Administratoren nur um die Verwaltung von Softwarepatches und Betriebssystemupdates kümmern.

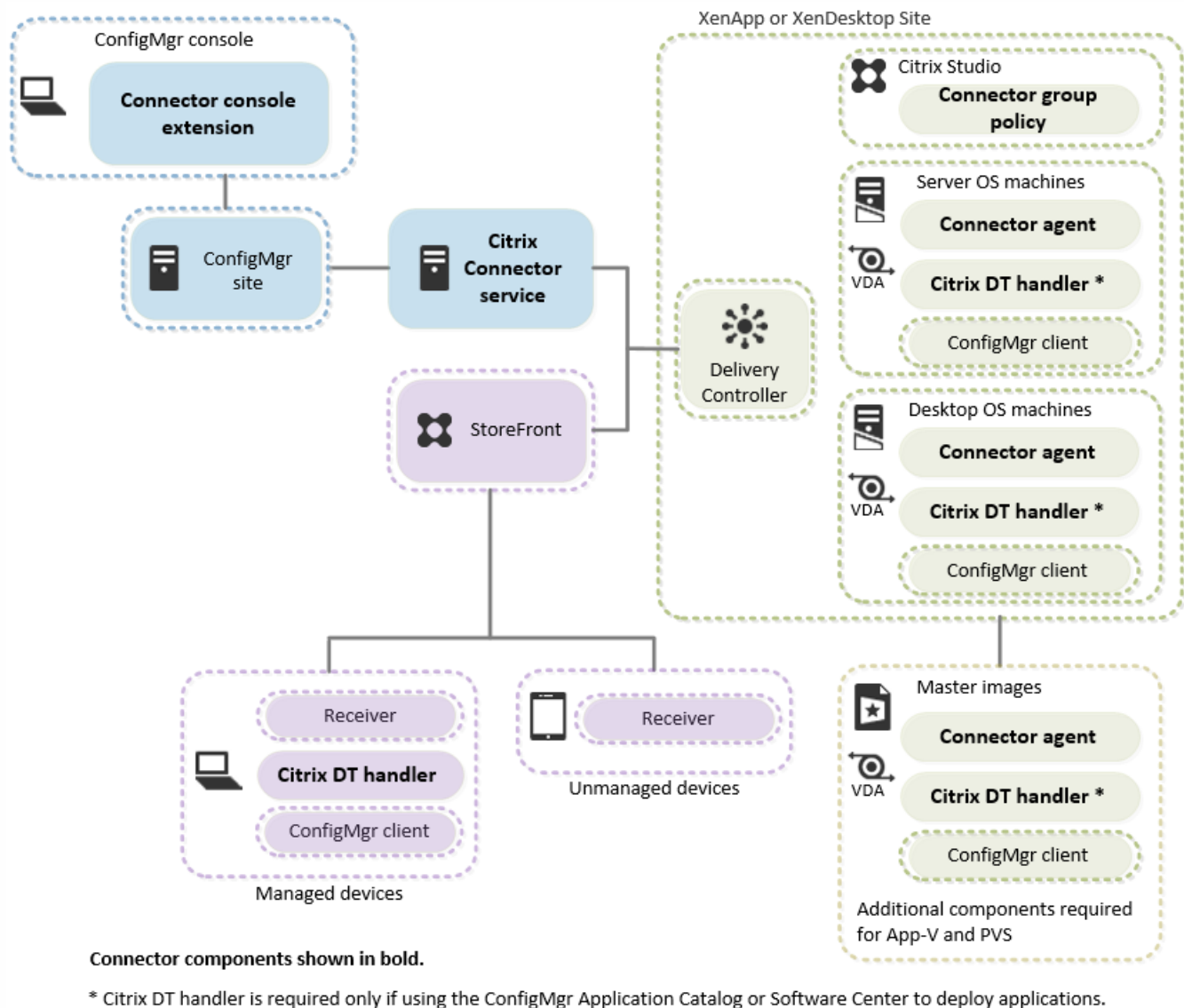
Die folgende Abbildung zeigt die Beziehung zwischen bereitgestellten Elementen, Configuration Manager und XenApp bzw. XenDesktop.



Wichtig: In Configuration Manager verwendet der Connector ein Updategerät zur Darstellung eines Masterimages. Vor dem Bereitstellen von Anwendungen für die meisten mit Maschinenerstellungsdienste oder Provisioning Services verwalteten Maschinenkatalogtypen wählen Sie ein Updategerät für die Bereitstellung aus.

Komponenten von Citrix Connector

Die folgende Abbildung zeigt die Komponenten einer Citrix Connector-Lösung. In diesem Abschnitt werden die einzelnen Komponenten beschrieben.



Citrix Connector-Dienst

Der Citrix Connector-Dienst bildet eine Brücke zwischen einer XenApp- bzw. XenDesktop (Citrix)-Bereitstellungsite und Configuration Manager. Der Connector-Dienst hat folgende Aufgaben:

- Synchronisieren von XenApp- bzw. XenDesktop-Maschinenkatalogen mit Gerätesammlungen
- Synchronisieren von XenApp- bzw. XenDesktop-Bereitstellungsgruppen mit Benutzersammlungen
- Orchestrieren der Softwareinstallation für Gerätesammlungen
- Veröffentlichen von Anwendungen für Benutzer in Bereitstellungsgruppen
- Bereitstellen des Citrix XenApp 7.5- bzw. XenDesktop 7.5-Bereitstellungstyps für Benutzer in Benutzersammlungen
- Sicherstellen, dass Anwendungen erst veröffentlicht werden, wenn sie auf allen erforderlichen Maschinen von Configuration Manager installiert wurden

- Bereitstellen hoher Verfügbarkeit bei Installation mehrerer Connector-Dienste

Configuration Manager-Konsolenerweiterung

Die Configuration Manager-Konsolenerweiterung ermöglicht die nahtlose Zusammenarbeit der Configuration Manager-Konsole mit XenApp bzw. XenDesktop 7.6 oder 7.5 und mit XenApp 6.5. Vom Connector werden der Configuration Manager-Konsole Objekte hinzugefügt. Beispiele:

- Ein Citrix Bereitstellungssite-Knoten unter Bestand und Kompatibilität > Gerätesammlungen. Dieser Knoten umfasst die Citrix Maschinenkataloge.

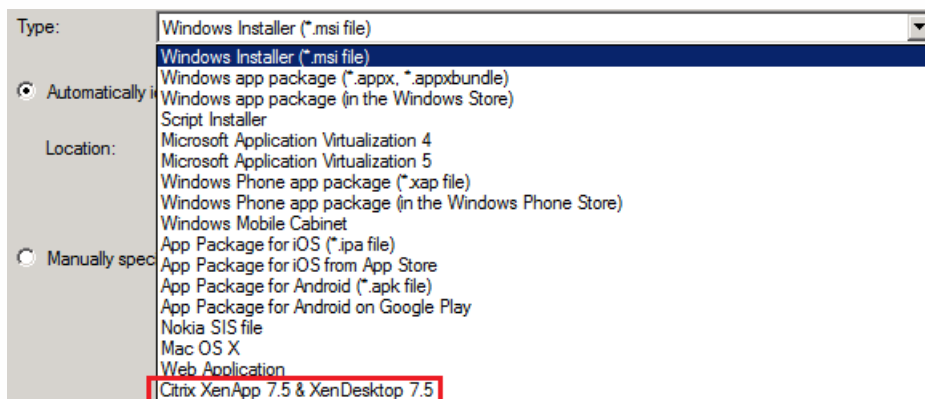


Der Connector fügt zudem einen Citrix Bereitstellungssite-Knoten unter Bestand und Kompatibilität > Benutzersammlungen hinzu. Dieser Knoten umfasst Citrix Bereitstellungsgruppen.

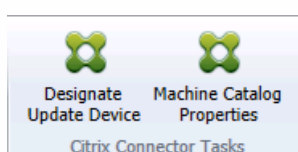
- Einen Knoten für Citrix Anwendungsveröffentlichungen unter Softwarebibliothek > Anwendungsverwaltung. Die Elemente in diesem Knoten werden auf allen Citrix Bereitstellungssites veröffentlicht.



- Ein Citrix-spezifischer Bereitstellungstyp, nämlich "Citrix XenApp 7.5" bzw. "XenDesktop 7.5", der nur für die Bereitstellung von durch Citrix gehosteten Anwendungen im Configuration Manager-Anwendungskatalog oder im Softwarecenter auf mit Configuration Manager verwalteten Geräten erforderlich ist.



- Zwei Befehle im Menüband von Configuration Manager, die angezeigt werden, wenn Sie einen Maschinenkatalog in einer Gerätesammlung auswählen.



Citrix Connector-Agent

Der Citrix Connector-Agent wird auf Desktopbetriebssystem- und Serverbetriebssystemmaschinen ausgeführt, die Mitglied von Citrix Maschinenkatalogen sind. Der Connector-Agent handhabt die Installation von Anwendungen und Softwareupdates in Koordination mit der Configuration Manager-Leerlaufrihtlinie. Der Connector-Agent orchestriert zudem Bereitstellungen für Serverbetriebssystemmaschinen, die manuell oder über Provisioning Services verwaltet werden.

Citrix Bereitstellungshandler

Der Citrix Bereitstellungshandler ist eine optionale Komponente, die nur zum Bereitstellen von über Citrix gehosteten Anwendungen für den Configuration Manager-Anwendungskatalog oder das Softwarecenter auf mit Configuration Manager verwalteten Geräten erforderlich ist. In diesem Szenario muss der Citrix XenApp 7.5- bzw. XenDesktop 7.5-Bereitstellungstyp die höchste Priorität haben.

Wichtig: Der Citrix Bereitstellungshandler ist zum Veröffentlichen von Anwendungen für Receiver nicht erforderlich. Der Citrix Bereitstellungshandler arbeitet mit dem Configuration Manager-Client wie folgt zusammen:

- Wenn bei einer Anwendung der Citrix XenApp 7.5- bzw. XenDesktop 7.5-Bereitstellungstyp die höchste Priorität hat, fügt der Citrix Bereitstellungshandler der Windows-Startseite bzw. dem Windows-Startmenü ein Symbol hinzu, das die von Citrix gehostete Anwendung startet. Für den Benutzer wirken solche Anwendungen in puncto Aussehen und Funktionsweise wie lokal installierte Anwendungen.
- Wenn eine Anwendung keinen Citrix XenApp 7.5- bzw. XenDesktop 7.5-Bereitstellungstyp hat oder dieser nicht die höchste Priorität hat, wird die Bereitstellung von Configuration Manager abgewickelt.

Citrix Richtlinien

Über Citrix Richtlinien wird festgelegt, wie der Connector-Agent Elemente wie Vorabwarnmeldungen oder Benachrichtigungen bezüglich erzwungener Abmeldung für Serverbetriebssystemmaschinen handhabt, die manuell oder über Provisioning Services verwaltet werden. Es gibt auch eine Richtlinie zum Konfigurieren der Wartungshäufigkeit des Connector-Agents.

Alle Connector-Richtlinien haben Standardeinstellungen. Prüfen Sie die Standardeinstellungen auf Eignung für Ihre Umgebung. Eine Beschreibung der Einstellungen sowie Informationen zu den Standardwerten finden Sie unter [Richtlinieneinstellungen für Connector für Configuration Manager](#).

XenApp- bzw. XenDesktop-VDA

Der XenApp- bzw. XenDesktop-VDA kommuniziert mit Delivery Controllern, die die Benutzerverbindungen verwalten. Mit dem VDA kann der Connector Informationen über ein Image abrufen, z. B. den Bereitstellungstyp, ob es sich um ein Masterimage handelt und ob das Betriebssystemimage veraltet ist.

Provisioning Services-Agent

Provisioning Services ermöglicht die Bereitstellung und die erneute Bereitstellung von Computern in Echtzeit von einem freigegebenen Datenträgerimage. Der auf einem Produktions-vDisk-Image ausgeführte Connector-Agent erkennt, wenn ein neues vDisk-Image zur Verfügung steht, und stellt das neue Image im nächsten Wartungsfenster bereit. Der Provisioning Services-Agent wird nur für freigegebene Images benötigt und muss auf dem vDisk-Masterimage installiert sein.

Citrix Receiver und StoreFront

Auf Geräten, die nicht mit Configuration Manager verwaltet werden, greifen Benutzer über mit StoreFront verwaltete Stores auf virtuelle Desktops und Anwendungen in Receiver zu. Der Connector funktioniert mit Receiver auf allen Benutzergeräten, die von XenApp bzw. XenDesktop unterstützt werden. Der Connector funktioniert auch mit der von XenApp bzw. XenDesktop unterstützten Webinterface-Version.

Auf mit Configuration Manager verwalteten Geräten können Benutzer auf Anwendungen über den Configuration Manager-Anwendungskatalog oder das Softwarecenter zugreifen. Obwohl die Benutzer Receiver in diesem Szenario nicht sehen, erfordert der Connector die Verwendung von StoreFront mit Receiver auf verwalteten Geräten. StoreFront liefert die Anwendungssymbole für den Configuration Manager-Anwendungskatalog und das Softwarecenter.

Systemanforderungen

Sep 29, 2015

Vergewissern Sie sich vor dem Installation des Connectors, dass Ihre Konfiguration den folgenden Anforderungen entspricht.

XenApp oder XenDesktop

Unterstützte Releases:

- XenApp 7.6 oder 7.5 (nur Platinum Edition)
- XenDesktop 7.6, 7.5 oder 7.1 (nur Platinum Edition)

Microsoft System Center 2012 Configuration Manager

Unterstützte Releases:

- System Center 2012 R2 SP1 Configuration Manager*
- System Center 2012 R2 Configuration Manager
- System Center 2012 SP2 Configuration Manager*
- System Center 2012 SP1 Configuration Manager

* Das Microsoft Hotfix muss angewendet werden. Weitere Informationen finden Sie unter <https://support.microsoft.com/en-us/kb/3074246>.

Citrix Connector-Dienst

Unterstützte Betriebssysteme:

- Windows Server 2012 R2-Editionen
- Windows Server 2008 R2-Editionen

Anforderungen:

- 50 MB Speicherplatz auf dem Datenträger für die Installation und zusätzlich bis zu 50 MB für die Protokollierung
- Verbindungen mit einem Citrix Delivery Controller
- Verbindungen mit der Configuration Manager-Standortserver
- .NET Framework 3.5 SP1 und .NET Framework 4.5.2, 4.5.1 oder 4.5

Citrix Bereitstellungshandler

Es muss einer der folgenden Clients installiert sein:

- System Center 2012 Configuration Manager R2-Client
- System Center 2012 Configuration Manager SP1-Client

Unterstützte Betriebssysteme:

- Windows Server 2012 R2-Editionen
- Windows Server 2008 R2-Editionen
- Windows 8,1
- Windows 8
- Windows 7 SP1

Connector-Agent

Unterstützte Betriebssysteme:

- Windows Server 2012 R2-Editionen
- Windows Server 2008 R2-Editionen
- Windows 8,1
- Windows 8
- Windows 7 SP1

Configuration Manager-Konsolenerweiterung

Unterstützte Releases:

- System Center 2012 Configuration Manager R2-Konsole
- System Center 2012 Configuration Manager SP1-Konsole

Unterstützte Betriebssysteme:

- Windows Server 2012 R2-Editionen
- Windows Server 2008 R2-Editionen
- Windows 8,1
- Windows 8
- Windows 7 SP1

Anforderungen:

- 50 MB Speicherplatz auf dem Datenträger für die Installation und zusätzlich bis zu 50 MB für die Protokollierung
- Verbindungen mit einem Citrix Delivery Controller
- Verbindungen mit der Configuration Manager-Standortserver
- .NET Framework 3.5 SP1 und .NET Framework 4.5.2, 4.5.1 oder 4.5

App-V-Client für System Center 2012 Configuration Manager R2-Konsole

Unterstützte Releases:

- App-V-Client für Remotedesktopdienste, Version 5.0 SP2
- App-V-Client für Remotedesktopdienste, Version 5.0 SP1
- App-V-Client für Remotedesktopdienste, Version 4.6.1 SP1

Active Directory

- Alle Komponenten müssen in der gleichen Domäne sein oder in Domänen, zwischen denen eine Vertrauensstellung besteht. Dies gilt für die XenApp- bzw. XenDesktop-Infrastruktur, die Configuration Manager-Infrastruktur, die Connector-Dienstmaschine, StoreFront und VM mit Masterimages.
- Benutzergeräte müssen in der gleichen Active Directory-Domäne sein wie die StoreFront-Stores. Wenn nicht verwaltete Benutzergeräte direkt auf einen XenApp- oder XenDesktop-Server verweisen, müssen sie nicht Mitglied einer Domäne sein.

Nicht verwaltete Benutzergeräte

Unter nicht verwalteten Geräten versteht man Geräte, auf denen der Configuration Manager-Client nicht installiert ist. Zu den nicht verwalteten Geräten gehören Mobilgeräte, Heimcomputer und Macintosh-Computer.

Anforderungen:

- Ein mit unterstützten Versionen von XenApp oder XenDesktop kompatibler Citrix Receiver

Der Connector unterstützt dieselben Receiver-Konfigurationen wie XenApp oder Desktop, einschließlich StoreFront und das Webinterface.

Verwaltete Benutzergeräte

Geräte, auf denen der Configuration Manager-Client installiert ist, gelten als verwaltete Geräte. Verwaltete Geräte müssen Mitglied einer Domäne sein.

Anforderungen:

- Configuration Manager-Client
- Citrix Bereitstellungshandler
- Citrix Receiver für Windows 4.1, 4.0 oder 3.4 (nur Standardedition)
Erforderlich für die Integration in Windows-Startseite bzw. -Menü. Den Benutzern wird die Receiver-Benutzeroberfläche nicht angezeigt. Dieses Feature wird von der Enterprise Edition von Receiver nicht unterstützt.
- StoreFront 2.5 oder 2.1
Erforderlich für die Integration in Anwendungskatalog und Software Center.

Weitere Informationen finden Sie unter [Konfigurieren der Integration von Windows-Startseite oder -Startmenü](#).

In StoreFront aggregierte Ressourcen

Anforderungen:

- StoreFront 2.5 oder 2.1
- Receiver für Windows 4.1
- Konfiguration von XenApp bzw. XenDesktop zur Veröffentlichung von Anwendungen aus mehreren Bereitstellungssites in einem einzelnen Store

Weitere Informationen zur Ressourcenaggregation in StoreFront finden Sie unter [Hohe Verfügbarkeit und Multisitekonfiguration für StoreFront](#).

Planen

Sep 29, 2015

Sie können mit einer einfachen Testbereitstellung für Citrix Connector beginnen und diese dann auf Folgendes erweitern:

- Hohe Verfügbarkeit
- Heterogene Umgebung mit XenApp 6.5 und XenApp 7.6 oder 7.5; heterogene Umgebung mit XenApp 6.5 und XenDesktop 7.6, 7.5 oder 7.1
- Mehrere XenApp 6.5-Farmen und XenApp-Bereitstellungssites (7.6 oder 7.5) oder XenDesktop-Bereitstellungssites (7.6, 7.5 oder 7.1)
- Mehrere geografische Bereiche
- Kombinationen der oben genannten Bereitstellungen

Für alle Bereitstellungen werden dieselben Komponenten verwendet. Lesen Sie vor dem Einrichten einer Bereitstellung die Informationen zu den Systemanforderungen.

Lesen Sie folgende Abschnitte mit allgemeinen Informationen zur Planung einer Bereitstellungsarchitektur:

[Machbarkeitsstudie](#)

[Heterogene Umgebungen mit Citrix Connector und XenApp Connector](#)

[Heterogene Umgebung mit hoher Verfügbarkeit](#)

[Unternehmensumgebung](#)

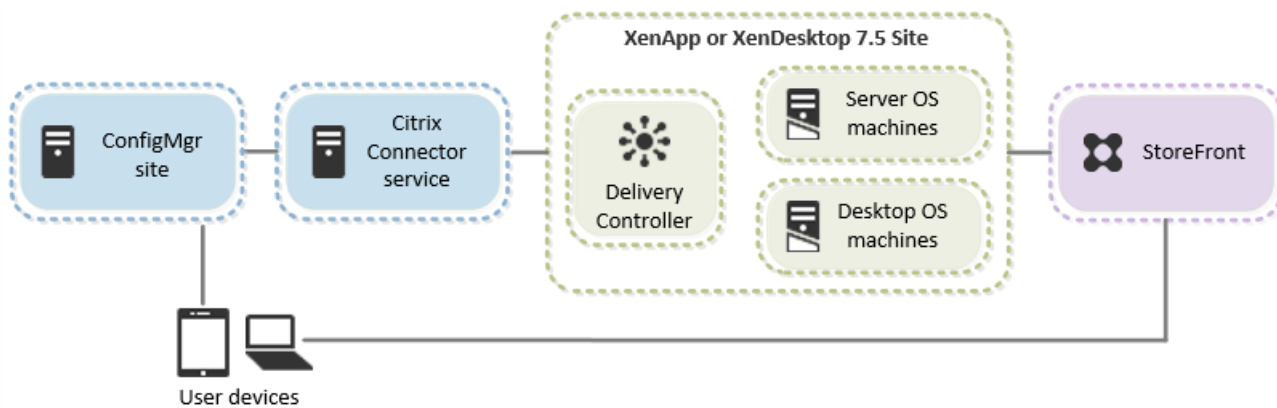
[Betrieb an mehreren geografischen Standorten](#)

Informationen zur Planung einer Verwendung von Configuration Manager mit Citrix Connector finden Sie unter [Strategien und bewährte Methoden](#).

Informationen zur Planung der Verteilung von Verwaltungsaufgaben finden Sie unter [Rollen und Aufgaben der Administratoren](#).

Machbarkeitsstudie

Die folgende Abbildung einer Testbereitstellung enthält einen Configuration Manager-Standort, eine Citrix Connector-Dienstmaschine und eine Citrix Bereitstellungssite. Bei einer Nicht-Produktionsumgebung darf der Citrix Connector-Dienst wie abgebildet auf einer separaten VM residieren oder auf dem Configuration Manager-Standortserver.



Der Citrix Connector-Dienst kommuniziert mit dem SMS-Anbieter von Configuration Manager und dem Citrix Delivery Controller.

Eine schrittweise Anleitung zu dem gesamten Verfahren, von der Installation über die Bereitstellung bis zur Veröffentlichung von Anwendungen, finden Sie im [Citrix Connector 7.5 Proof of Concept Deployment Guide](#).

Heterogene Umgebungen mit Citrix Connector und XenApp Connector

Die Bereitstellung darf Citrix Connector 7.5 und XenApp Connector 6.5 enthalten, sofern es sich um eine der folgenden heterogenen Umgebungen handelt:

- XenApp 6.5 und XenApp 7.6 oder 7.5
- XenApp 6.5 und XenDesktop 7.6, 7.5 oder 7.1

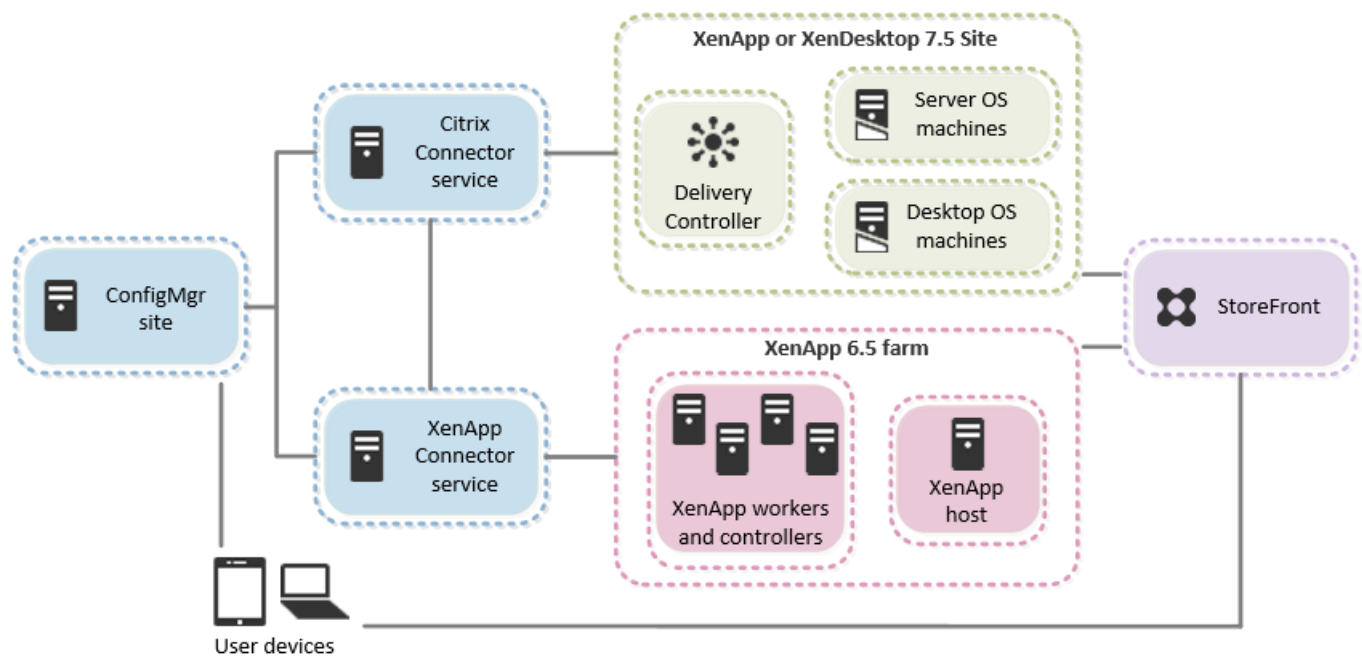
In einer heterogenen Umgebung wird vom Citrix Connector eine einheitliche Liste der gehosteten Anwendungen aus XenApp 6.5-Farmen und XenApp- oder XenDesktop-Bereitstellungssites angezeigt. Sie können für die heterogene Umgebung Wartungsfenster konfigurieren und Anwendungen von beiden Umgebungen aus bereitstellen und veröffentlichen.

Die Installationsprogramme der Komponenten des Citrix Connectors helfen Ihnen, die richtige Kombination aus Komponenten in einer heterogenen Umgebung zu installieren. Wenn Sie beispielsweise versuchen, den Connector 7.5-Agent auf einer Maschine mit dem XenApp 6.5 Connector-Agent zu installieren, wird diese Aktion vom Installationsprogramm als ungültig gemeldet und unterbunden.

Das folgende Bereitstellungsdiagramm zeigt eine heterogene Umgebung mit einer XenApp- bzw. XenDesktop-Bereitstellungssite und einer XenApp 6.5-Farm. Eine Citrix Connector-Dienstmaschine und eine XenApp Connector-Dienstmaschine verweisen auf eine einzelne Configuration Manager-Site.

Die beiden Connectors können auf der gleichen VM oder auf separaten VMs residieren.

Es wird empfohlen, den Connector-Dienst in einer Produktionsumgebung nicht auf dem Configuration Manager-Standortserver zu installieren.



Der Citrix Connector-Dienst und der XenApp Connector-Dienst kommunizieren mit dem SMS-Anbieter von Configuration Manager. Der Citrix Connector-Dienst kommuniziert mit dem Citrix Delivery Controller. Der XenApp Connector-Dienst kommuniziert mit einem XenApp-Host, bei dem es sich um einen Controller (keine Workermaschine) handeln muss.

Heterogene Umgebung mit hoher Verfügbarkeit

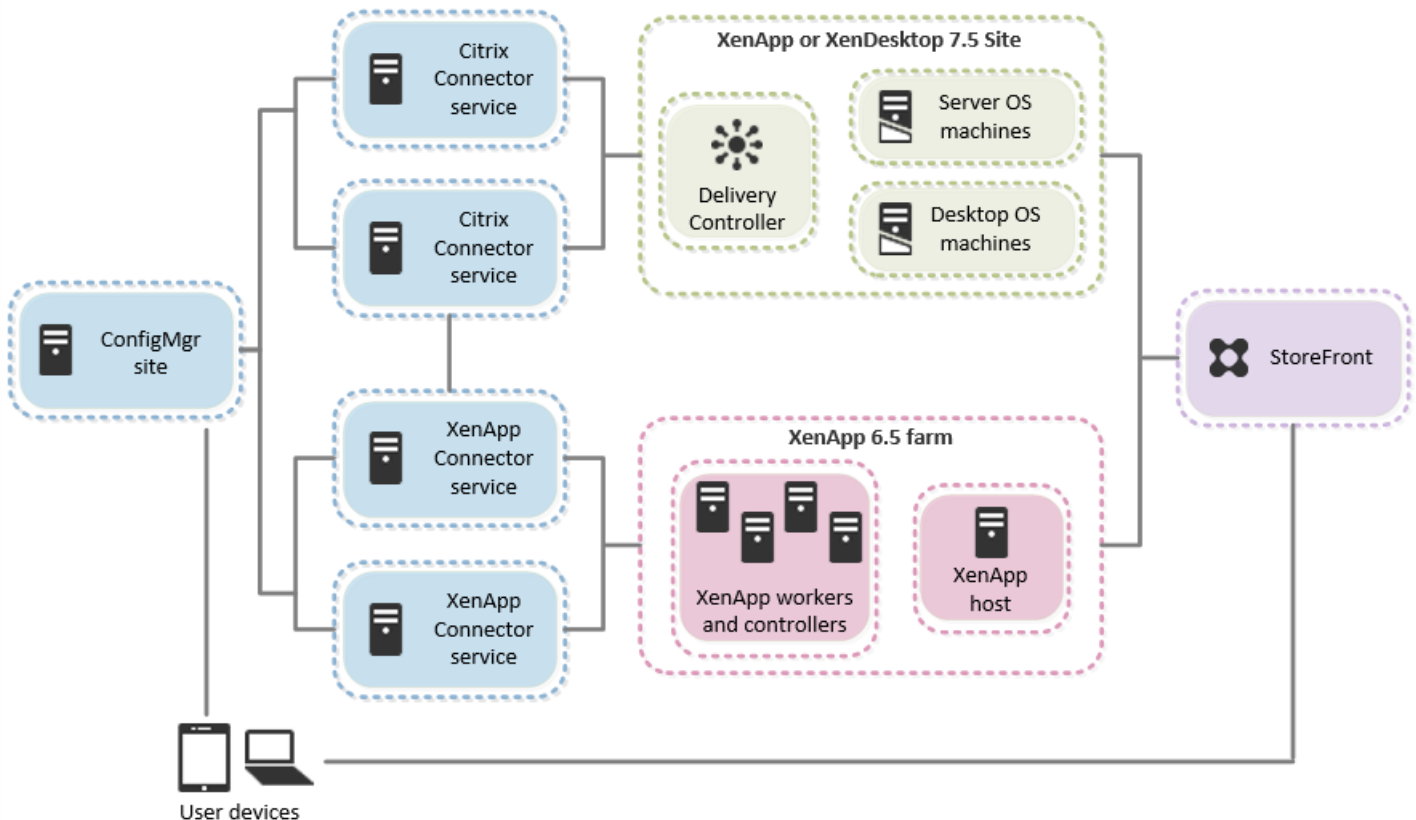
Das Feature für hohe Verfügbarkeit des Connectors stellt eine zuverlässige Fehlertoleranzmethode bereit, um Dienstkontinuität bei Ausfall von Infrastrukturkomponenten (Software, Hardware, Netzwerk und Strom) sicherzustellen.

Eine Bereitstellung mit hoher Verfügbarkeit enthält mehrere Maschinen mit dem Connector-Dienst. Der Connector verwendet optional ein Aktiv/Passivmodell für hohe Verfügbarkeit. Nur ein Connector-Dienst wird jeweils pro XenApp- bzw. XenDesktop-Bereitstellungssite ausgeführt; dies minimiert die Ressourcennutzung und garantiert Betriebskontinuität. Wenn der aktive Connector-Dienst betriebsunfähig wird, übernimmt automatisch ein anderer Connector-Dienst seine Rolle. Das Hinzufügen von Maschinen mit dem Connector-Dienst erhöht nicht die Kapazität. Eine Bereitstellung mit hoher Verfügbarkeit wird daher unabhängig von der Größe der Organisation empfohlen.

Die aktive Instanz und zugehörige Informationen werden aus Persistenzgründen in der Configuration Manager-Datenbank gespeichert

Das folgende Bereitstellungsdiagramm zeigt eine heterogene Umgebung mit einer XenApp- bzw. XenDesktop-Bereitstellungssite und einer XenApp 6.5-Farm. Zwei Citrix Connector-Dienstmaschinen bzw. zwei XenApp Connector-Dienstmaschinen gewährleisten hohe Verfügbarkeit in ihrer Bereitstellungssite bzw. Farm.

Die beiden Connectors können auf der gleichen VM oder auf separaten VMs residieren. Es wird empfohlen, den Connector-Dienst in einer Produktionsumgebung nicht auf dem Configuration Manager-Standortserver zu installieren.



Der Citrix Connector-Dienst und der XenApp Connector-Dienst kommunizieren mit dem SMS-Anbieter von Configuration Manager. Eine beliebige Anzahl Connector-Dienste kann auf einen Configuration Manager-Standortserver verweisen. Unterschiedliche Connector-Dienste können auf unterschiedliche SMS-Anbieter an einem Configuration Manager-Standort verweisen. Dadurch wird die Fehlertoleranz erhöht, denn es wird das Entstehen einer einzelnen Fehlerquelle auf Seiten von Configuration Manager vermieden.

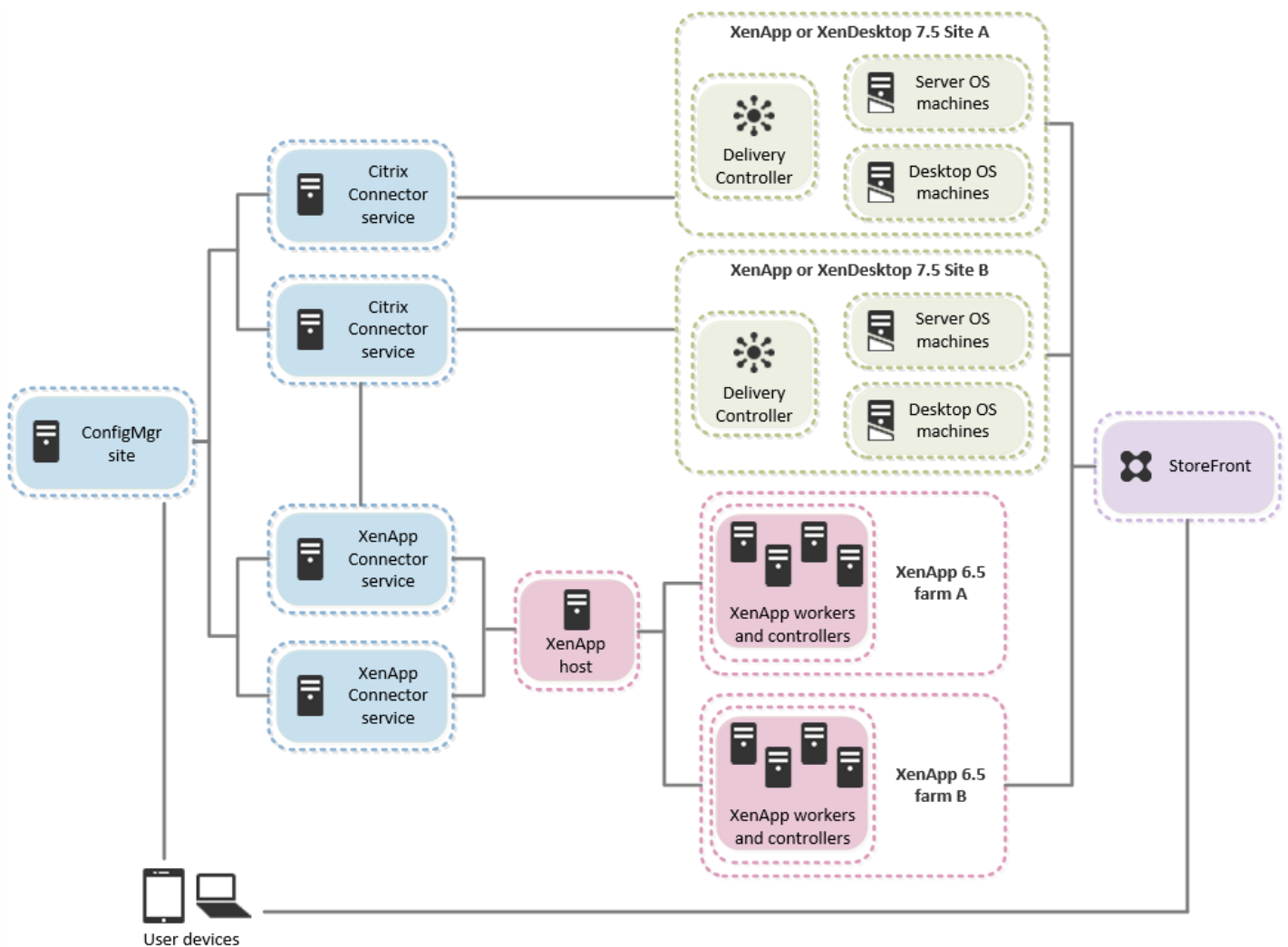
Die Configuration Manager-Standortdatenbank wird auch zur Speicherung der Tabelle für hohe Verfügbarkeit verwendet. Diese enthält Informationen dazu, welcher Connector-Dienst zu einem gegebenen Zeitpunkt für eine bestimmte Bereitstellungsite bzw. Farm aktiv ist.

Der Citrix Connector-Dienst kommuniziert mit dem Citrix Delivery Controller. Der XenApp Connector-Dienst kommuniziert mit einem XenApp-Host, bei dem es sich um einen Controller (keine Workermaschine) handeln muss. Bei einer Bereitstellungsite mit mehreren Delivery Controllern wird durch den Verweis mehrerer Connectors auf einen Delivery Controller keine hohe Verfügbarkeit erzielt, wenn der Delivery Controller betriebsunfähig wird. Verwenden Sie daher mindestens einen Connector-Dienst pro Delivery Controller.

Unternehmensumgebung

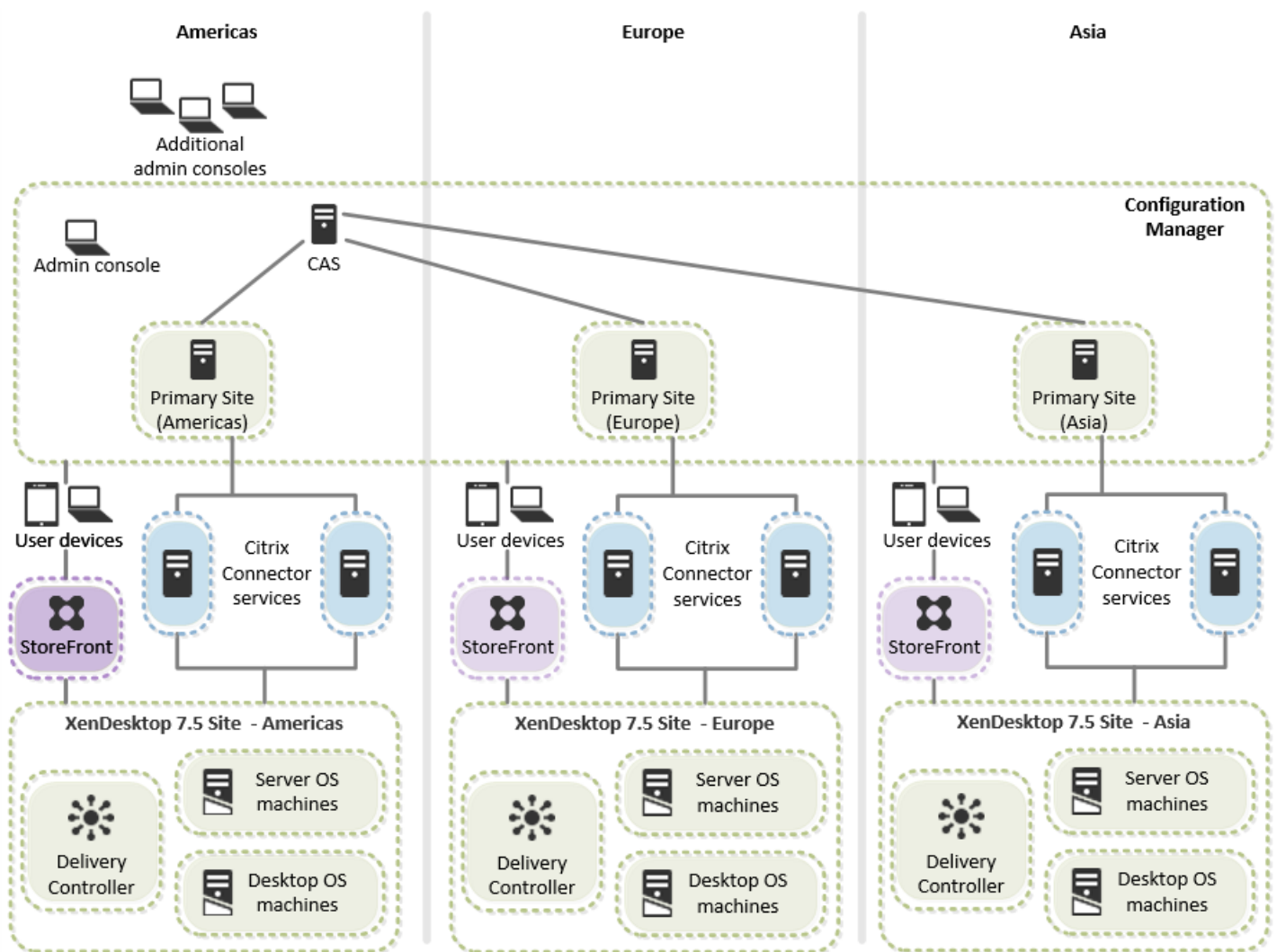
Die einfachste Methode der Verwaltung mehrerer Citrix Bereitstellungssites mit Citrix Connector besteht in der Einrichtung mindestens einer Citrix Connector-Dienstmaschine für jede Site. Sie konfigurieren jeden Connector-Dienst unabhängig, als ob die Site, auf die er verweist, die einzige im Unternehmen wäre. Citrix Connector verarbeitet automatisch das Vorhandensein der anderen Citrix Bereitstellungssites und funktioniert ohne Konflikt.

Was alles andere betrifft, zeigt das folgende Bereitstellungsdiagramm die gleiche Einrichtung als heterogene Umgebung mit hoher Verfügbarkeit.



Betrieb an mehreren geografischen Standorten

Das folgende Beispiel einer Bereitstellung zeigt eine große Sitehierarchie mit mehreren geografischen Standorten, in der ein Standort der zentralen Verwaltung von Configuration Manager mit drei primären Standorten (Amerika, Europa und Asien) verwendet wird.



Bei der Planung der Bereitstellung von Citrix Connector in einer spezifischen Configuration Manager-Topologie ist Folgendes zu beachten:

- Platzieren Sie die verschiedenen Citrix Connector-Dienstmaschinen immer in Netzwerknähe zu den Citrix Bereitstellungssites und den Configuration Manager-Standortservern.
- Lassen Sie zu, dass Configuration Manager die siteübergreifende Kommunikation, die Replikation, langsame Verbindungen usw. abwickelt.
- Vermeiden Sie soweit wie möglich lange Kommunikationswege zwischen dem Citrix Connector-Dienst und den zugewiesenen Maschinen.

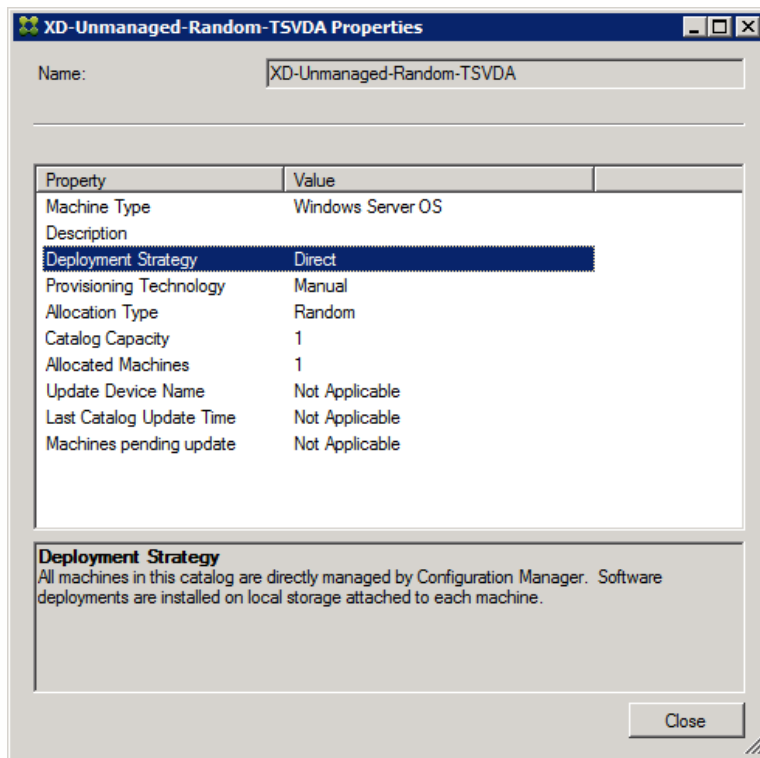
Sekundäre, nicht in der Abbildung dargestellte Standorte, werden durch den jeweils übergeordneten primären Standort und somit von dem oder den Citrix Connector-Dienst(en) im zugehörigen primären Standort verwaltet. Das Installieren von Citrix Connector auf sekundären Standortservern ist nicht sinnvoll und wird nicht empfohlen.

Strategien und bewährte Methoden

Gerätesammlungen und Maschinenkataloge

- -
-
-
-

-
-



Wartung von Gerätesammlungen

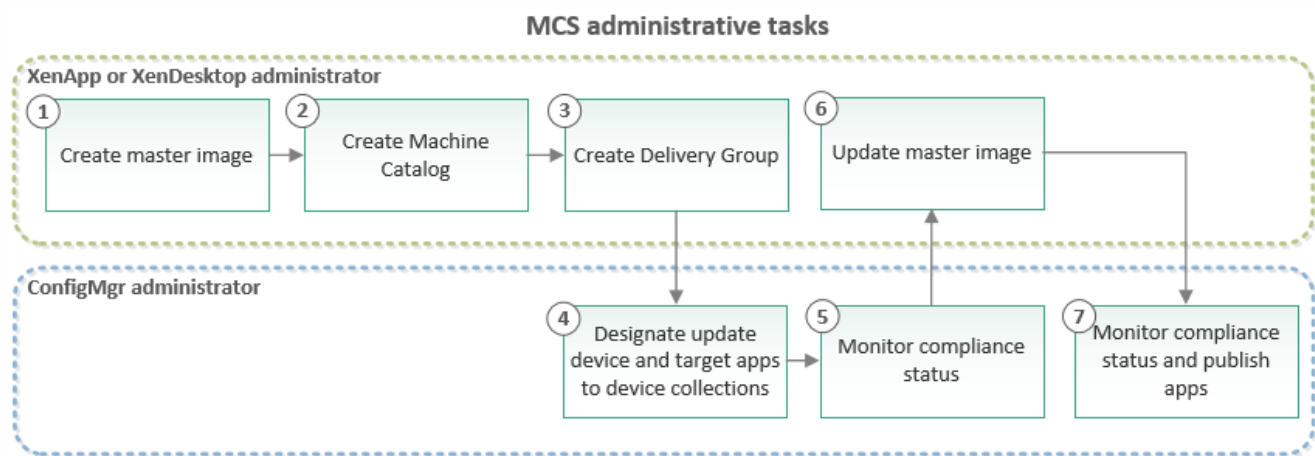
-

-

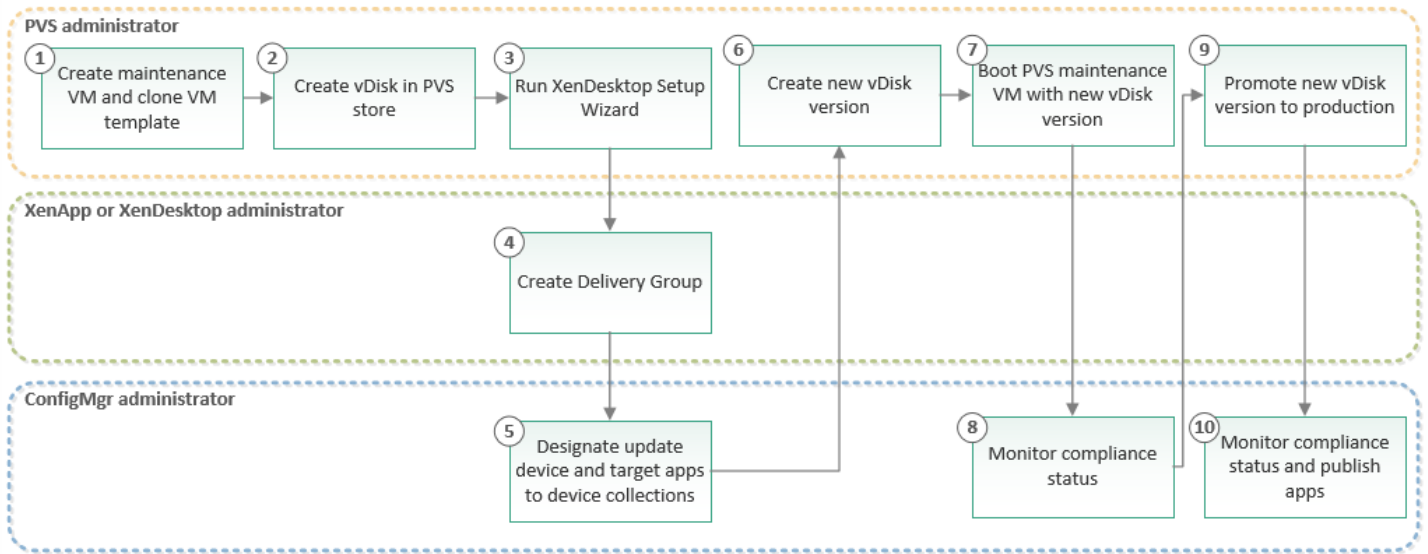
Anwendungsbereitstellungstypen

-

-



PVS administrative tasks



Bereitstellungsorchestrierung

-
-
-

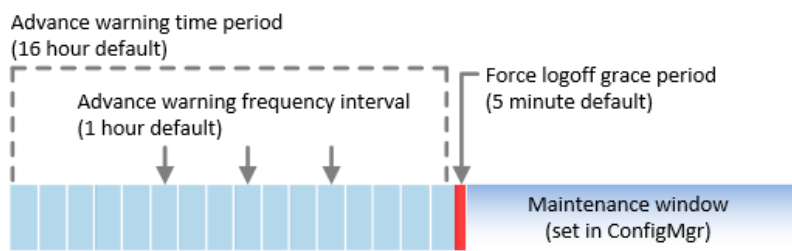
Wartungsfenster

-
-
-
-

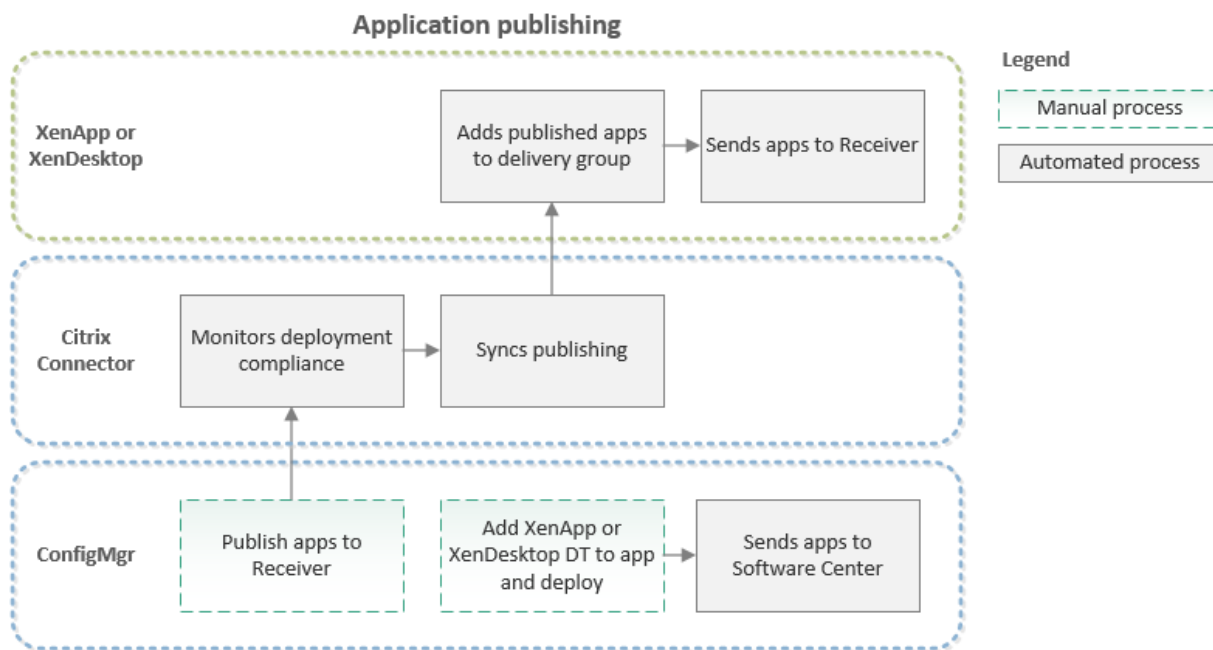
Leerlaufrichtlinie von Configuration Manager

-
-

Benutzerbenachrichtigungen



Anwendungsveröffentlichung

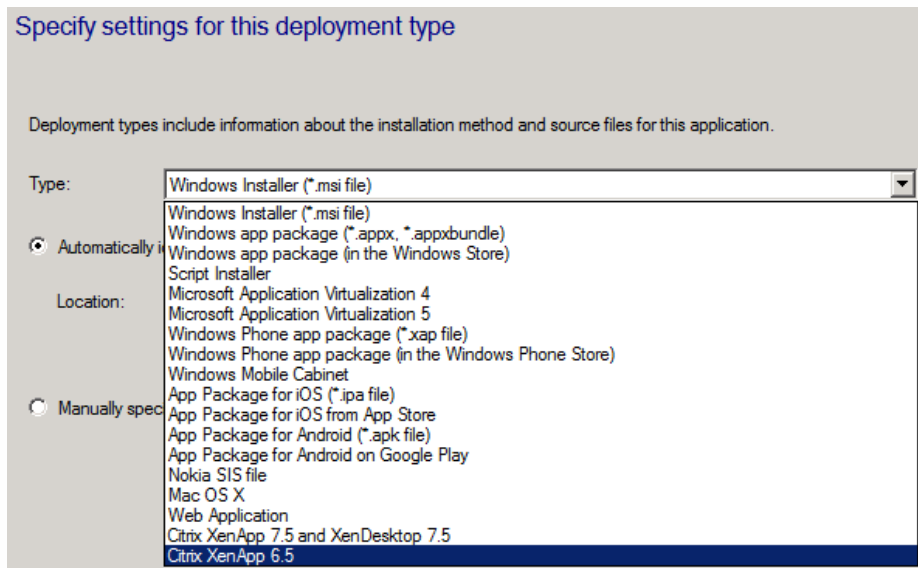
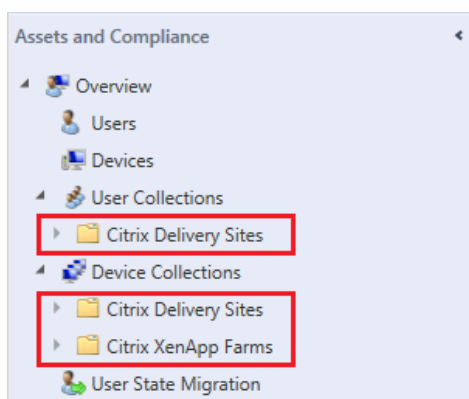


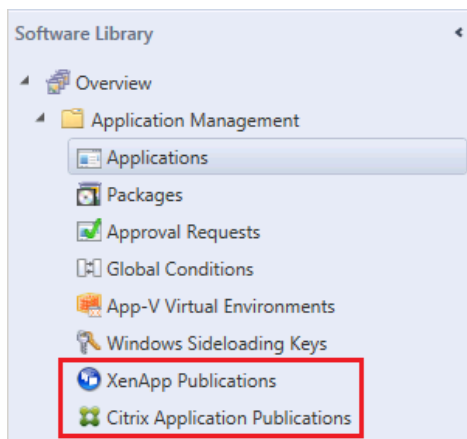
Anwendungsveröffentlichung und Benutzererfahrung

-
-
-
-
-
-
-
-

Bereitstellungen mit dem XenApp 6.5-Connector

-





•

•

•

•

Rollen und Aufgaben der Administratoren

•

•

•

•

•

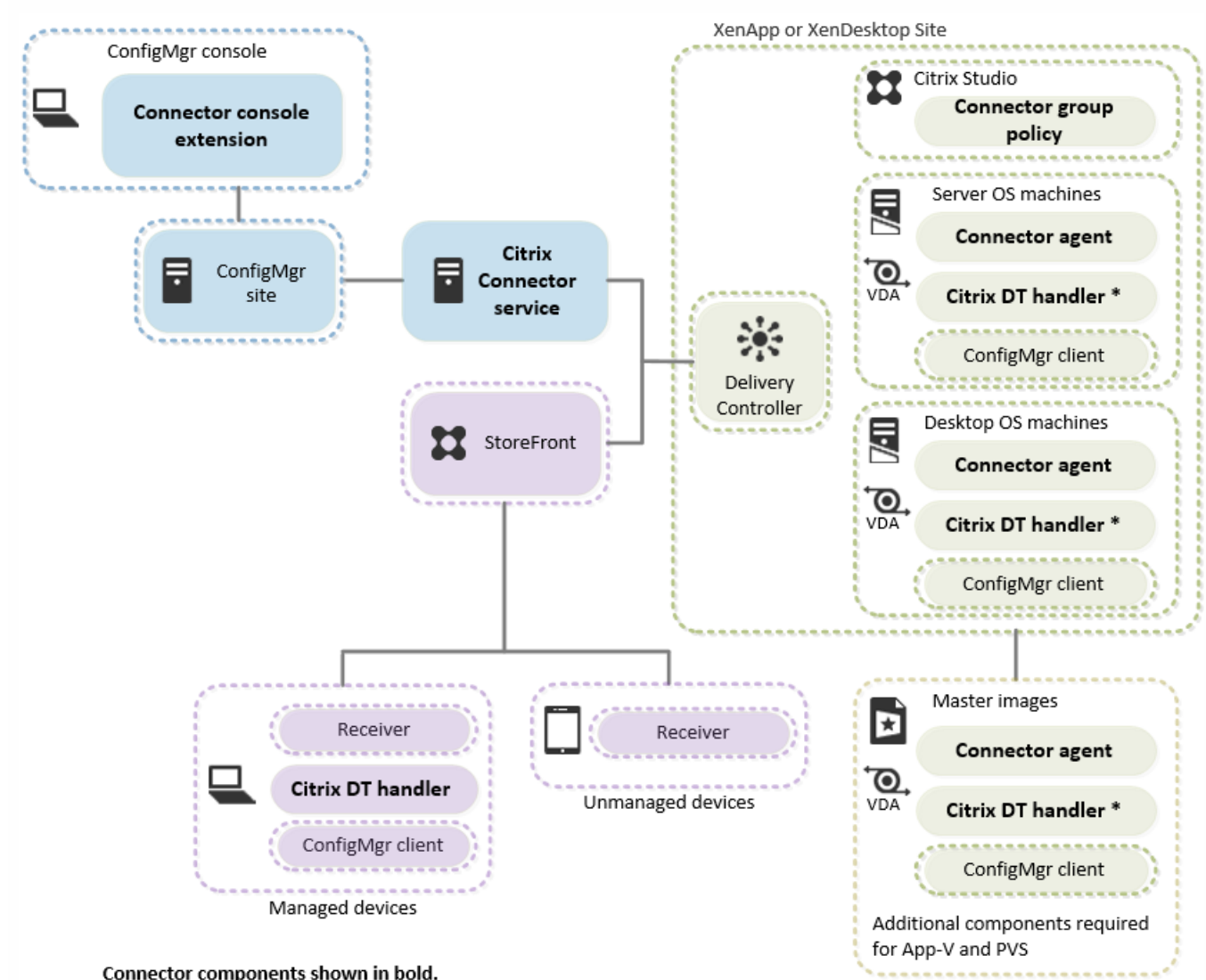
-
-
-
-
-
-
-
-

-
-
-
-
-
-
-

-
-

-
-

Installieren von Citrix Connector



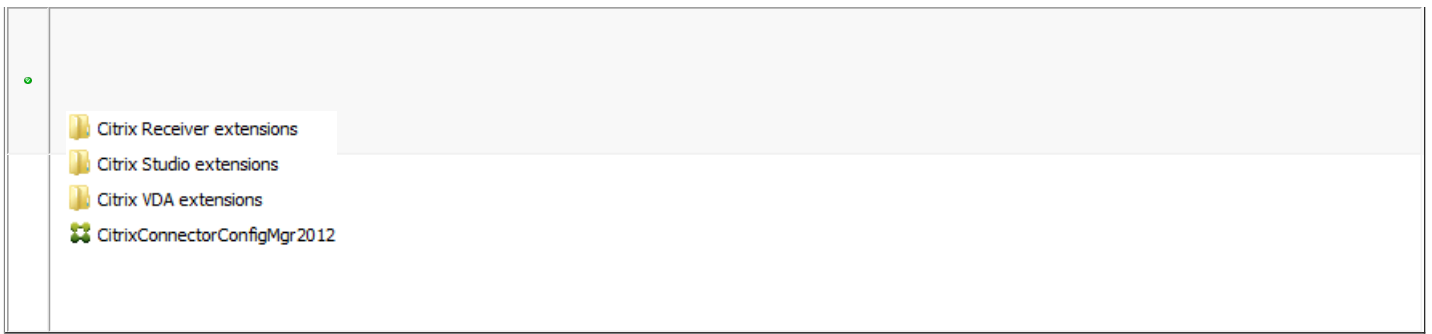
Connector components shown in bold.

* Citrix DT handler is required only if using the ConfigMgr Application Catalog or Software Center to deploy applications.

•
•

Vorbereiten der Installation

	
	• • • • • • • • • •
	Winrm set winrm/config/winrs '@{MaxMemoryPerShellMB="1024"}' Set-Location wsman:\localhost\Shell Get-ChildItem



Installieren des Citrix Connector-Diensts und der Konsolenerweiterung

Installieren des Connector-Diensts und der Konsolenerweiterung

Installieren der Connector-Konsolenerweiterung auf weiteren Maschinen

Konfigurieren des Connectors

-
-

-
-

-

-

Installieren von Komponenten auf Masterimages oder Sitzungsmaschinen

-

-

-

-

-

-

-
-

-
-

Installieren von Connectorrichtlinien

-
-

Installieren von Komponenten auf Benutzergeräten

Nicht verwaltete Benutzergeräte

Verwaltete Benutzergeräte

-
-
-

-
-
-

-
-

Unbeaufsichtigte Installation der Connector-Komponenten

Connector-Agent-Befehlszeilenoptionen

	CitrixConnectorAgent_x64[86].msi /help
	CitrixConnectorAgent_x64[86].msi /openfwport /quiet
	CitrixConnectorAgent_x64[86].msi /uninstall

Befehlszeilenoptionen des Citrix Bereitstellungshandlers

	CitrixDTHandler_x64[86].msi /help
	CitrixDTHandler_x64[86].msi /quiet
	CitrixDTHandler_x64[86].msi /uninstall

Upgrade

Sichern und Wiederherstellen

Exportieren einer Konfigurationsdatei

%Programme%\Citrix\Connector for ConfigMgr\Config Wizard

	Citrix.ConfigMgr.ConfigWizard.exe /help
	Citrix.ConfigMgr.ConfigWizard.exe /E[xportFile]
	Citrix.ConfigMgr.ConfigWizard.exe /E[xportFile]=ConfigDataToday.xml
	Citrix.ConfigMgr.ConfigWizard.exe /E[xportFile]=C:\ConfigDataToday.xml

Connector_Service_Account" Password="" />

Exportieren einer Connector-Konfigurationsdatei

%Programme%\Citrix\Connector for ConfigMgr\Config Wizard

	Citrix.ConfigMgr.ConfigWizard.exe /I[importFile] /A[pplly]
	Citrix.ConfigMgr.ConfigWizard.exe /I[importFile]=D:\ConfigDataToday.xml /A[pplly]

Deinstallieren des Connectors

`%ProgramFiles%\Citrix\Connector for ConfigMgr\Config Wizard\Citrix.ConfigMgr.ConfigWizard.exe /U[ninstall]`

-
-
-

Vorgänge bei der Deinstallation des Connectors

-
-
-
-

Erstellen von Anwendungen

Bereits in Configuration Manager vorhandene Anwendungen

Erstellen einer MSI- oder App-V-Anwendung

-
-

-

Erstellen einer skriptbasierten Anwendung

-

-

Erstellen von Citrix Receiver-Anwendungen

-
-

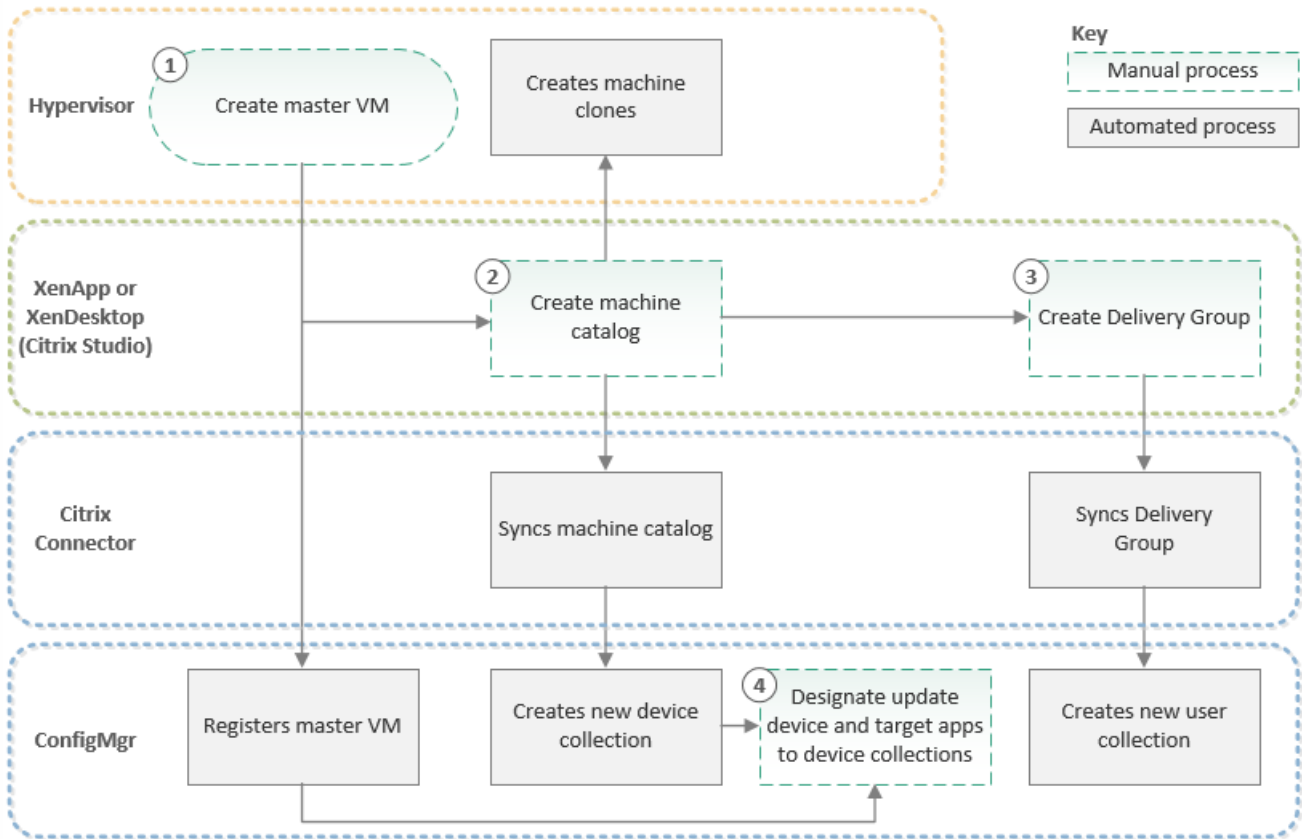
Bereitstellen von Anwendungen für Maschinenkataloge

-
-
-

Von MCS verwaltete Maschinenkataloge

Vorbereiten der Anwendungsbereitstellung

MCS setup and configuration



-
-
-

-
-

Bereitstellen von Anwendungen auf von einem Updategerät verwalteten Desktopbetriebssystem- oder Serverbetriebssystem-Sitzungsmaschinen

-
-

Deploy Software Wizard



Deployment Settings

General

Content

Deployment Settings

Scheduling

User Experience

Alerts

Summary

Progress

Completion

Specify settings to control how this software is deployed

Action:

Install

Purpose:

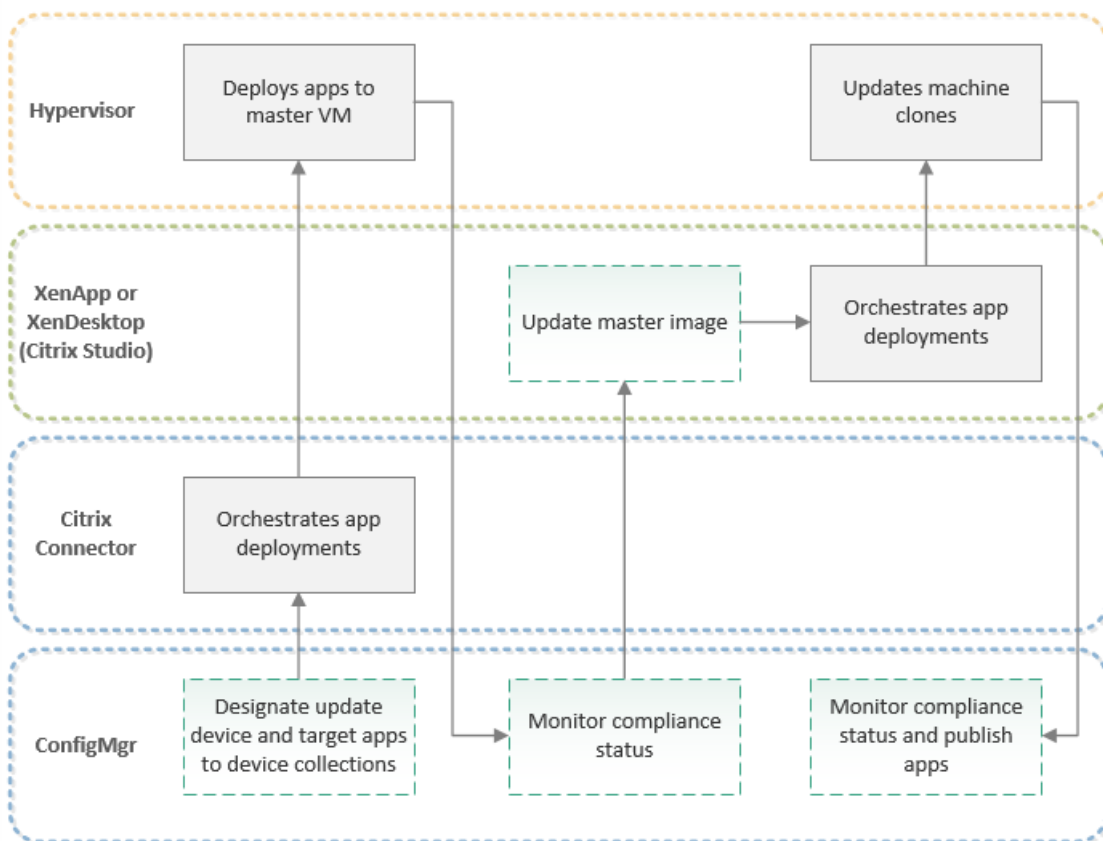
Required

☐ Pre-deploy software to the user's primary device

☐ Send wake-up packets

☐ Allow clients on a metered Internet connection to download content after the installation deadline, which might incur additional costs

MCS image maintenance



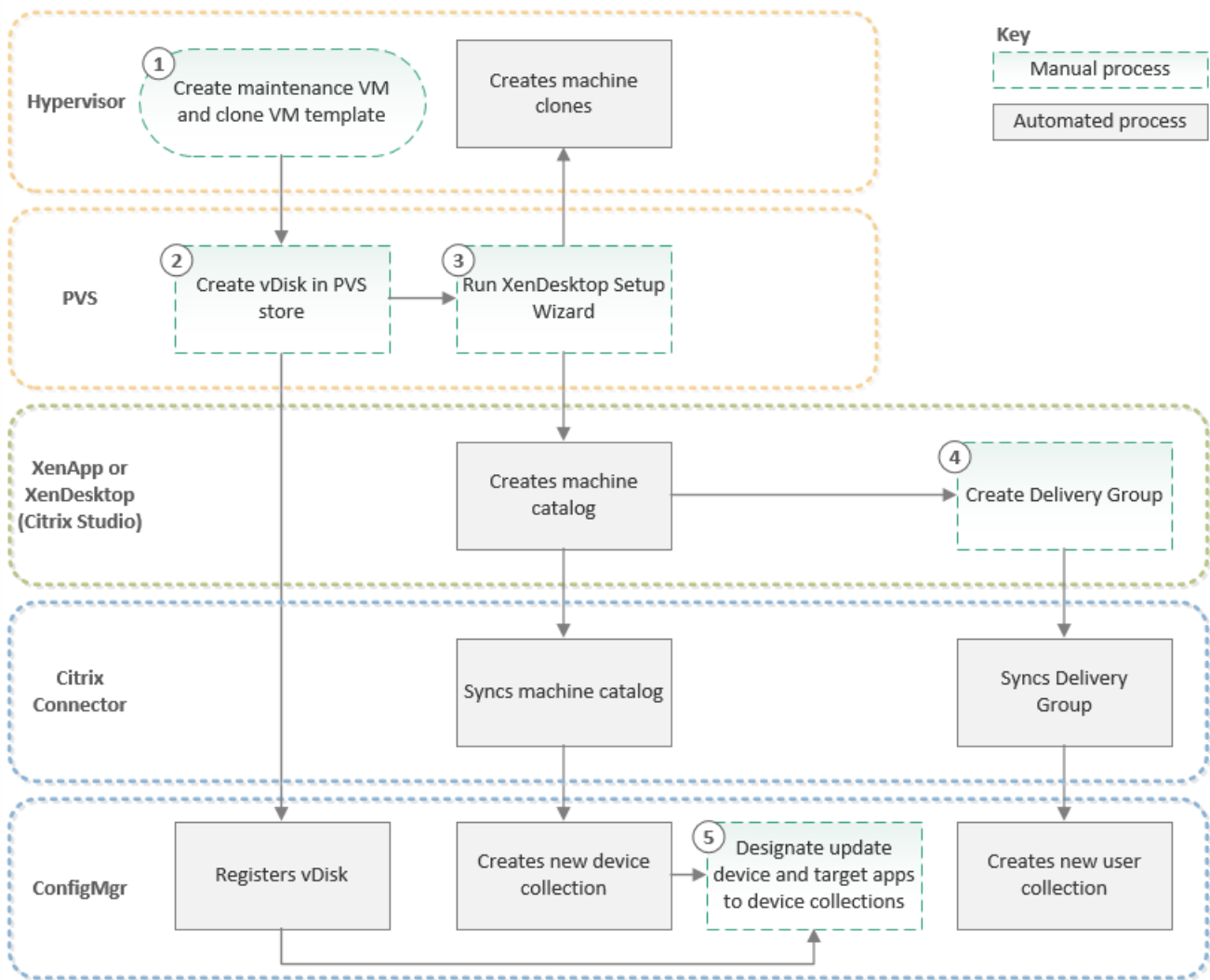
Bereitstellen von Anwendungen auf Desktopbetriebssystem-Sitzungsmaschinen mit Speicherung von Benutzerdaten auf einem lokalen Datenträger



Von Provisioning Services verwaltete Maschinenkataloge

Vorbereiten der Anwendungsbereitstellung

PVS setup and configuration



•

•

•

Bereitstellen von Anwendungen für ein Updategerät

-
-

Deploy Software Wizard

Deployment Settings

General

Content

Deployment Settings

Scheduling

User Experience

Alerts

Summary

Progress

Completion

Specify settings to control how this software is deployed

Action:

Install

Purpose:

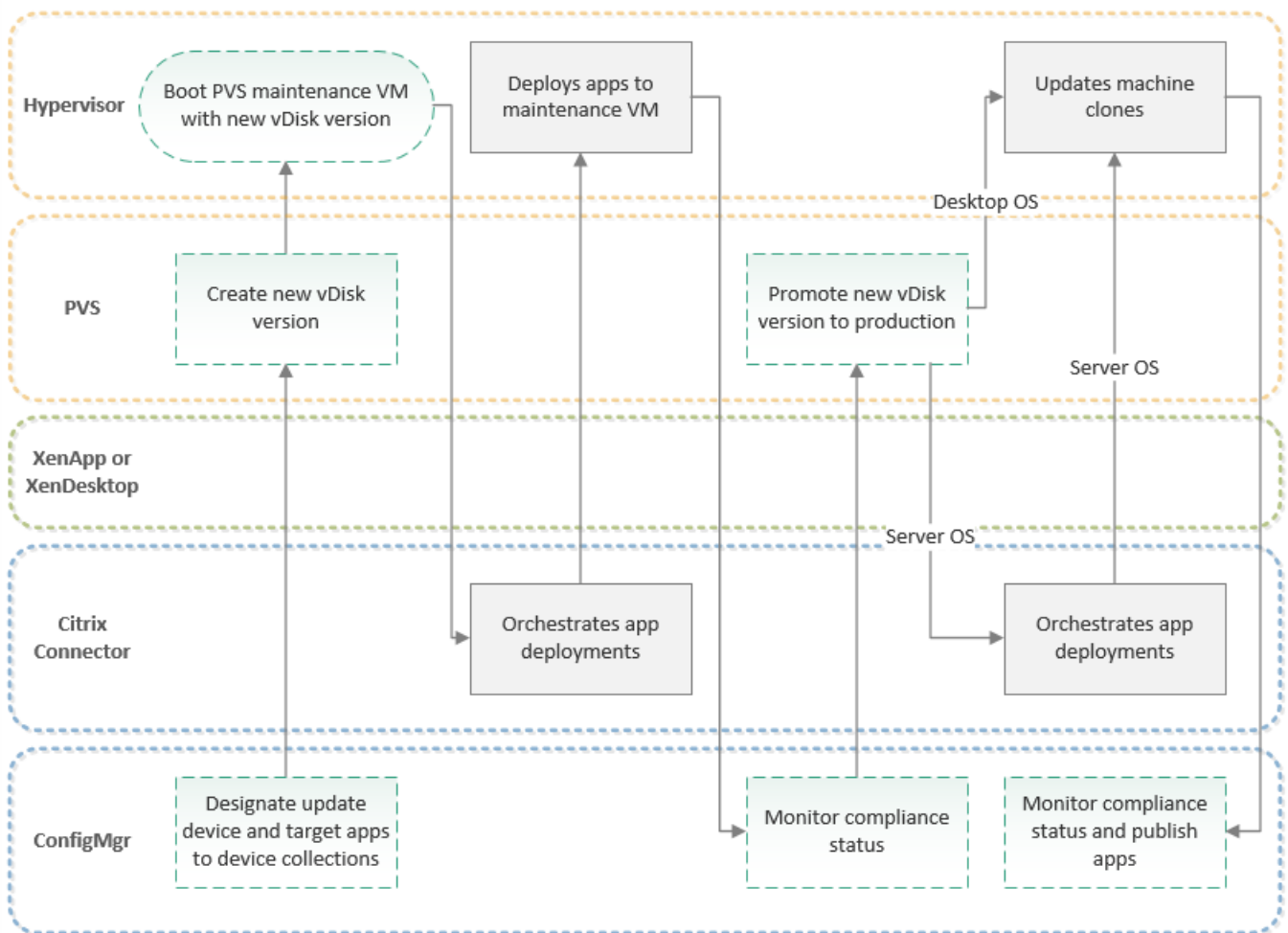
Required

☐ Pre-deploy software to the user's primary device

☐ Send wake-up packets

☐ Allow clients on a metered Internet connection to download content after the installation deadline, which might incur additional costs

PVS image maintenance



Manuell verwaltete Maschinenkataloge

-
-

Deploy Software Wizard

Deployment Settings

General

Content

Deployment Settings

Scheduling

User Experience

Alerts

Summary

Progress

Completion

Specify settings to control how this software is deployed

Action:

Install

Purpose:

Required

☐ Pre-deploy software to the user's primary device

☐ Send wake-up packets

☐ Allow clients on a metered Internet connection to download content after the installation deadline, which might incur additional costs



Veröffentlichen von Anwendungen

-
-
-

Veröffentlichen von Anwendungen für Receiver

-
-
-

Überprüfen der Veröffentlichung

-

Konfigurieren der Integration von Windows-Startseite oder -Startmenü

-

-

-

-

-

-

-

-

-

-

-

Ändern der Einstellung zum Melden des Installations-/Deinstallationszustands

-

-

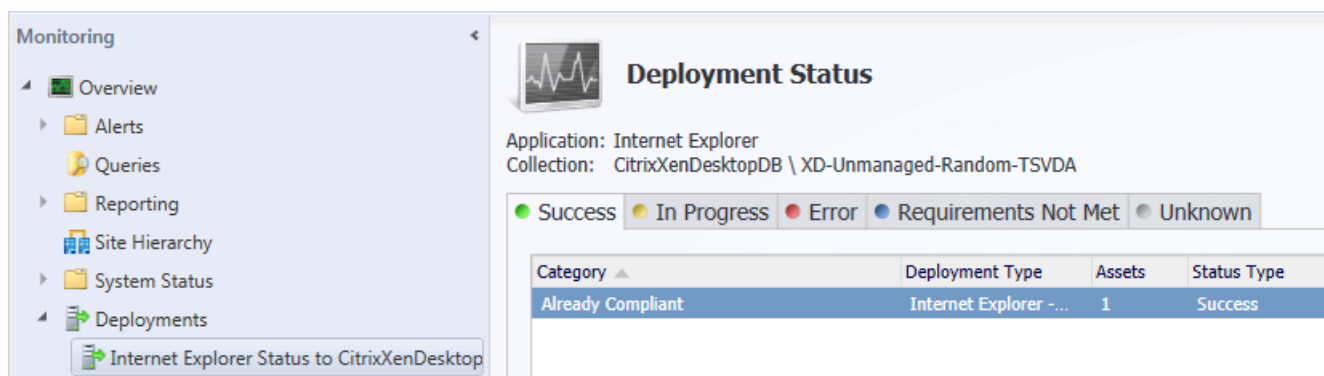
Rationalisieren der Bereitstellung obligatorischer, mit dem Citrix Bereitstellungstyp veröffentlichter Anwendungen

Überwachen und Problembehandlung

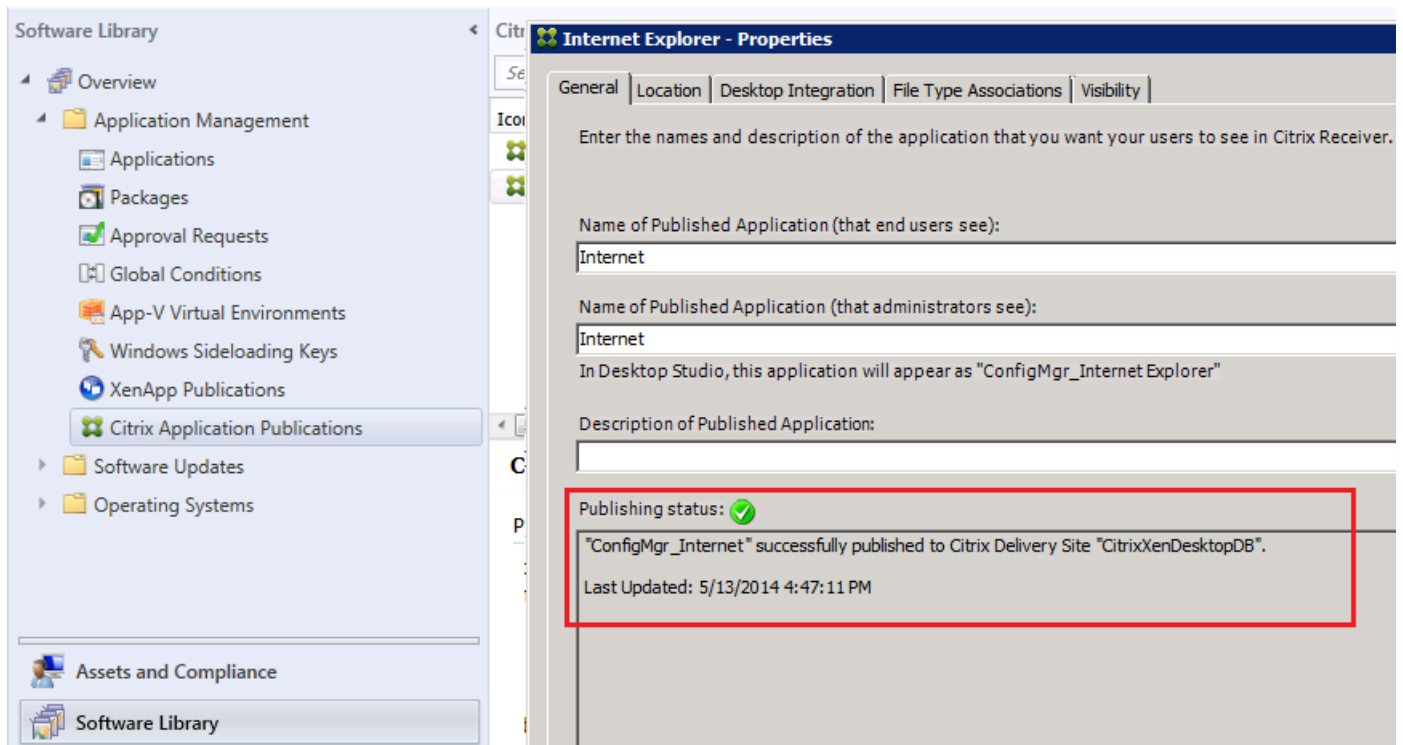
-
-
-

-
-
-

Überwachen der Anwendungsbereitstellung



Überwachen der Veröffentlichung von Anwendungen



Konfigurieren der Connector-Protokollierung

-
-

-

•

baseFilename	
enabled	
appendMode	
sizeLimit	
trailCount	

Verwenden von Protokolldateien für die Problembehandlung

Problembehandlung

	• •

	• •

Lizenzierung

Neue Funktionen

Behobene Probleme

-

Bekannte Probleme

-
-
-
-
-
-

Installationsprobleme

-

-

-

-

-

-

Sonstige bekannte Probleme

-

-

-

-

-

-

Simple License Service

-

-

-

-

-

Lizenzdateien

-

-

-

-

-

-

Ereignisprotokollmeldungen

-

-

Produktspezifische Informationen

-

-

-

-

-

Systemanforderungen für die Citrix Lizenzierung

Anforderungen für die Lizenzierung für Windows

	• • • • • • •
	• •
	• • • •

Verwalten der Lizenzierung

-
-

Editionen und unterstützte Lizenzmodelle

	• • •	
	• • •	• •

Anzeigen von Lizenzierungsinformationen

Herunterladen einer Lizenz von Citrix

Hinzufügen von Lizenzen von einem Netzwerk

Ändern des Lizenzservers

Auswählen des zu verwendenden Lizenztyps

Ändern der Produktedition und des Lizenzierungsmodells

Zugreifen auf die License Administration Console

Hinzufügen eines Lizenzadministrators

Bearbeiten oder Löschen eines Lizenzadministrators

Hinzufügen einer Lizenzadministratorgruppe

Bearbeiten oder Löschen einer Lizenzadministratorgruppe

Verbindungen und Ressourcen

Erstellen einer Verbindung und Ressourcen

-
-
- -
 -
 -
- -
 -
 -
- -
 -
 -

-

-

Erstellen einer Verbindung und Ressourcen aus einer vorhandenen Verbindung

Hinzufügen einer Verbindung zu einem Speicher

Bearbeiten einer Verbindung

-

-

-

-

-
-
-
-
-

-
-

Versetzen einer Verbindung in den Wartungsmodus

Löschen einer Verbindung

-
-
-
-

Umbenennen einer Verbindung

Anzeigen von Maschinendetails

Verwalten von Maschinen

-

-

-

-

-

-

-

-

Löschen von Ressourcen

Umbenennen von Ressourcen

Verwenden von IntelliCache

•
•
•
•
•

•
•

•
•

Update von Benutzerdesktops

Hinzufügen von Maschinen

Ändern von Maschinenkatalogen

-
-
-
-
-
-
-
-
-
-

Hinzufügen von Maschinen im Maschinenkatalog

Vorbereitung

-

-
-
-
-
-

Hinzufügen von Maschinen zum Maschinenkatalog

-
-
-
-
-
-
-

Ändern von Maschinenkatalogen

-
-
-
-
-

Bearbeiten von Maschinenkatalogen

-
-

Umbenennen von Maschinenkatalogen

Suchen nach Maschinenkatalogen

Löschen von Maschinenkatalogen und enthaltenen Maschinen

-
-
-
-
-

Löschen eines Maschinenkatalogs

-
-
-

Löschen von Maschinen aus einem Maschinenkatalog

-
-
-
-
-

Verwalten von Desktopcomputerkonten

-

-

Verwalten von Active Directory-Konten

-

-

-

-

-

-

-

-

Verwalten von Organisationseinheiten

-

-

Aktualisieren eines Masterimages

-

-

Aktualisieren des Masterimages

-

-

-
-

Zurücksetzen des Masterimages auf die vorherige Version

Aktualisieren eines Maschinenkatalogs

-
-
-

-
-

-
-

Aktualisieren von Maschinenkatalogen

-
-

-
-
-
-

Rückgängigmachen eines Maschinenkatalogupgrades

Verwalten der Anwendungs- und Desktopbereitstellung

Bearbeiten von Bereitstellungsgruppen

	• • •
	• •

Entfernen der Maschinenkatalogzuordnung einer Remote-PC-Zugriff-Bereitstellungsgruppe

Suchen nach einer Bereitstellungsgruppe

Erstellen einer Bereitstellungsgruppenanwendung

-

-

-

-

-

-

-

-

Erstellen von Anwendungen für eine Bereitstellungsgruppe

-

-

-

-

-

-

-

Ändern und Verwalten von Anwendungen

Ändern der Benutzereinstellungen für Bereitstellungsgruppen

Suchen nach Maschinen, Sitzungen, Maschinenkatalogen, Anwendungen und Bereitstellungsgruppen



-
-
-
-

Einstellen der Energieverwaltung der Desktopbetriebssystemmaschinen

-

-

-

-

-

-

-

-

-

Importieren und Exportieren von Benutzerdaten

Sep 29, 2015

Sie können den Benutzern Desktops und Anwendungen zuweisen, indem Sie Daten aus einer CSV-Datei importieren, vorausgesetzt:

- Die Bereitstellungsgruppe basiert auf vorhandenen oder physischen Maschinen.
- Sie haben die erforderlichen Berechtigungen für den Zugriff auf die Datei und die Bereitstellungsgruppe

Diese CSV-Datei kann Daten aus einer vorherigen Version enthalten und Sie können Sie nur verwenden, um Bereitstellungsgruppen zu aktualisieren, die auf physischen Maschinen basieren.

Sie können auch Benutzerdaten in eine Datei exportieren.

Import- und Exportdateien müssen folgende Eigenschaften aufweisen:

- Sie müssen CSV-Dateien sein.
- Die erste Zeile in der Datei muss eine der folgenden Spaltenüberschriften enthalten:
 - [ADComputerAccount],[AssignedUser],[VirtualMachine],[HostId].
 - Die Spaltenüberschriften können in beliebiger Reihenfolge sein, müssen jedoch durch Trennzeichen getrennt sein.
- Die folgenden Zeilen enthalten die entsprechenden, durch Trennzeichen getrennten Daten: Die ADComputerAccount-Einträge können Folgendes sein:
 - Allgemeine Namen (z. B. Computer01)
 - IP-Adressen (z. B. 10.50.10.80)
 - Distinguished Name (z. B. Computer01.meineDomäne.com)
 - Domänen- und Computernamenpaare (z. B. meineDomäne\Computer01)

1. Wählen Sie in Studio unter Bereitstellungsgruppen die Bereitstellungsgruppe aus, deren Daten importiert oder exportiert werden sollen.
2. Klicken Sie auf Bereitstellungsgruppe bearbeiten.
3. Klicken Sie auf der Seite Zuweisung bearbeiten auf Importieren oder Exportieren.

Verwalten von Maschinensitzungen

Sep 29, 2015

Wenn sich ein Benutzer bei einer Maschine anmeldet, wird vom Benutzergerät eine Verbindung mit dem Virtual Delivery Agent (VDA) auf der Maschine hergestellt und eine Sitzung eingerichtet. Beim Ausführen von Wartungs- oder Hilfsmaßnahmen für Benutzer können Sie die Sitzungen wie folgt steuern:

- Abmelden von Benutzern von Sitzungen
- Trennen von Sitzungen
- Senden von Nachrichten an Benutzer
- Suchen nach Sitzungen, Benutzern und Maschinen

Je nach Maschinentyp können Sie Sitzungen abmelden und trennen.

- Wenn Sie eine Sitzung abmelden, wird sie geschlossen und die Maschine steht anderen Benutzern zur Verfügung, sofern sie nicht einem bestimmten Benutzer zugewiesen ist.
- Wenn Sie eine Sitzung trennen, werden die Anwendungen des Benutzers weiter ausgeführt und die Maschine bleibt dem Benutzer zugewiesen. Wenn der Benutzer die Verbindung wiederherstellt, wird dieselbe Maschine zugewiesen.

Hinweis: Abhängig vom Maschinentyp der Sitzung können Sie Energiestatustimer konfigurieren, um nicht genutzte Sitzungen automatisch zu verarbeiten. Dadurch werden Desktops freigegeben und es wird Energie gespart. Beispiel: Sie können das System so einrichten, dass getrennte Sitzungen automatisch nach 10 Minuten abgemeldet werden.

1. Verwenden Sie in Studio die Option Suchen, um nach der Sitzung zu suchen, oder wählen Sie eine Desktopgruppe aus und klicken Sie auf Maschinen anzeigen.
2. Wählen Sie die Sitzung oder den Desktop aus und klicken Sie auf Abmelden oder Trennen.

Sie können Nachrichten an Benutzer senden, um sie auf Maschinenwartungen hinzuweisen, die sich auf ihre Sitzungen auswirken. So können Sie beispielsweise Benutzern mitteilen, dass sie sich abmelden sollen, bevor wichtige Wartungsaufgaben durchgeführt werden.

1. Verwenden Sie in Studio die Option Suchen, um nach der Sitzung, der Maschine oder dem Benutzer zu suchen. Wählen Sie alternativ eine Desktopgruppe aus und klicken Sie auf Maschinen anzeigen.
2. Wählen Sie die Sitzung, den Desktop oder den Benutzer aus und klicken Sie auf Nachricht senden.
3. Erstellen Sie die Nachricht.

Aktivieren oder Deaktivieren des Wartungsmodus

Sep 29, 2015

Wenn Sie vorübergehend alle Verbindungen einer Maschine trennen möchten, um Wartungsaufgaben durchzuführen, setzen Sie die Maschine in den Wartungsmodus. Sie können Bereitstellungsgruppen oder einzelne Maschinen in den Wartungsmodus setzen.

Wenn Sie eine Maschine in den Wartungsmodus setzen, können Sie Verwaltungsaufgaben auf dem zugeordneten Image durchführen, zum Beispiel Patches oder Upgrades mit Image-Verwaltungstools anwenden.

Der Wartungsmodus wirkt sich wie folgt auf die Verbindungen der Benutzer aus:

- Bei Serverbetriebssystemmaschinen können Benutzer eine Verbindung mit vorhandenen Sitzungen herstellen, aber keine neuen Sitzungen starten.
- Bei Desktopbetriebssystemmaschinen und Remote-PC-Access-Maschinen können Benutzer keine Verbindung herstellen, wenn die Maschine im Wartungsmodus ist. Besteht bereits eine Verbindung, bleibt diese erhalten, bis sie getrennt wird oder eine Abmeldung erfolgt.

Maschinen werden für Benutzerverbindungen verfügbar, sobald der Wartungsmodus beendet wird.

1. Verwenden Sie die Suche in Studio, oder wählen Sie eine Bereitstellungsgruppe und klicken Sie auf Maschinen anzeigen, um einzelne Maschinen auffindig zu machen. Sie können auch den Knoten Bereitstellungsgruppen wählen, um eine Bereitstellungsgruppe anzuzeigen und zu finden.
2. Wählen Sie die Maschine oder die Bereitstellungsgruppe aus und klicken Sie auf Wartungsmodus aktivieren oder Wartungsmodus deaktivieren.

Verwalten von Bereitstellungsgruppen-Ressourcen

Sep 29, 2015

Wenn Sie eine Maschine aus einer Bereitstellungsgruppe löschen, wird die zugehörige virtuelle Maschine nicht aus dem Maschinenkatalog gelöscht, auf dem die Gruppe basiert. Die Maschinen sind daher für Zuweisungen zu anderen Bereitstellungsgruppen weiter verfügbar.

Sie können Maschinen nur entfernen, wenn sie im Wartungsmodus sind und heruntergefahren werden. Wenn Sie vorübergehend verhindern möchten, dass Benutzer eine Verbindung mit der Maschine herstellen, während Sie sie löschen, setzen Sie die Maschine in den [Wartungsmodus](#), bevor Sie sie herunterfahren.

Wichtig: Maschinen enthalten möglicherweise persönliche Daten. Bedenken Sie dies, insbesondere wenn eine Maschine einem anderen Benutzer zugewiesen wird. Sie müssen beispielsweise u. U. ein neues Image einer virtuellen Maschine erstellen.

1. Verwenden Sie in Studio die Option Suchen, um nach der Maschine zu suchen, oder wählen Sie eine Bereitstellungsgruppe aus und klicken Sie auf Maschinen anzeigen.
2. Wählen Sie die Maschine aus und setzen Sie sie in den Wartungsmodus.
3. Stellen Sie sicher, dass die Maschine heruntergefahren ist.
4. Klicken Sie auf Aus Bereitstellungsgruppe entfernen.

Sie können Maschinen in Bereitstellungsgruppen für alle Maschinentypen, mit Ausnahme der mit Provisioning Services erstellten, neu zuweisen.

Die Desktopzuweisung ist bei Serverbetriebssystemmaschinen nicht verfügbar.

1. Wählen Sie in Studio den Knoten Bereitstellungsgruppen.
2. Wählen Sie die Bereitstellungsgruppe mit den Maschinen aus, die Sie neu zuweisen möchten und wählen Sie dann Bereitstellungsgruppe bearbeiten.
3. Fügen Sie auf der Seite Maschinenzuweisung die Benutzer und Gruppen hinzu, die auf alle Maschinen in der Gruppe zugreifen können, oder entfernen Sie sie. Sie können auch eine Importliste verwenden, um die Benutzer und Gruppen anzugeben, die auf alle vorhandenen oder physischen Maschinen in einer Gruppe zugreifen können. Weitere Informationen finden Sie unter [Importieren und Exportieren von Benutzerdaten](#).

1. Wählen Sie in Studio den Knoten Bereitstellungsgruppen.
2. Wählen Sie die Bereitstellungsgruppe mit den Maschinen aus, die Sie neu zuweisen möchten, und wählen Sie dann Maschinen anzeigen.
3. Wählen Sie die gewünschte Maschine aus und wählen Sie Benutzer ändern.
4. Fügen Sie auf der Seite Benutzerzuweisung bearbeiten die Benutzer und Gruppen hinzu, die auf die Maschine in der Gruppe zugreifen können, oder entfernen Sie sie.

Tags sind Zeichenfolgen, mit denen die Maschinen gekennzeichnet werden. Anhand dieser Tags können Sie eine Maschinensuche verfeinern oder den Zugriff auf Maschinen beschränken. Sie können eine beliebige Anzahl von Tags mit

beliebiger Länge hinzufügen. Tags wie Vertrieb, Marketing oder Test kennzeichnen beispielsweise bestimmte Maschinen für verschiedene Benutzer.

Sie können Tags nur für Desktopbetriebssystemmaschinen verwenden.

1. Wählen Sie eine Maschine aus. In Studio:
 - Wählen Sie den Knoten Bereitstellungsgruppen, wählen Sie eine Desktopbetriebssystem-Bereitstellungsgruppe und klicken Sie dann auf Maschinen anzeigen.
 - Wählen Sie Suchen, um die Maschine zu finden.
2. Wählen Sie eine Maschine aus und wählen Sie dann Tag hinzufügen.
3. Geben Sie ein oder mehrere Tags durch Semikolons (;) getrennt ein und klicken Sie dann auf Tag hinzufügen.
4. Wählen Sie zum Ändern oder Entfernen von Tags eine Maschine aus (wie in Schritt 1 beschrieben) und klicken Sie dann auf Tag bearbeiten.
5. Ändern Sie die Tags und klicken Sie auf OK.

Sie können die Anzahl zugewiesener Maschinen für Benutzer einer Bereitstellungsgruppe vergrößern oder verkleinern.

1. Wählen Sie in Studio den Knoten Bereitstellungsgruppen.
2. Wählen Sie die Bereitstellungsgruppe der Benutzer, denen mehr oder weniger Maschinen zugewiesen werden sollen, klicken Sie auf Bereitstellungsgruppe bearbeiten und dann auf Endbenutzereinstellungen.
3. Legen Sie die Anzahl der Maschinen pro Benutzer auf der Seite Endbenutzereinstellungen fest.

1. Wählen Sie in Studio den Knoten Bereitstellungsgruppen.
2. Wählen Sie die Bereitstellungsgruppe mit den Maschinen aus, die Sie aktualisieren möchten und wählen Sie dann Maschinen anzeigen.
3. Wählen Sie die gewünschte Maschine aus, klicken Sie auf Maschinen aktualisieren und dann auf Masterimage.
4. Wählen Sie einen Snapshot aus der Liste aus. Wählen Sie einen Snapshot aus und klicken Sie auf das Pluszeichen, um die zugehörigen Masterimages anzuzeigen.

1. Wählen Sie in Studio den Knoten Bereitstellungsgruppen.
2. Wählen Sie die Bereitstellungsgruppe mit den Maschinen aus, die Sie aktualisieren möchten und wählen Sie dann Maschinen anzeigen.
3. Wählen Sie die gewünschte Maschine aus, klicken Sie auf Maschinen aktualisieren und dann auf Rollout und Benachrichtigung.
4. Wenn Sie Änderungen am Masterimage vorgenommen haben, legen Sie fest, wann die Aktualisierung des Masterimages erfolgen soll:
 - Sofort (Maschine jetzt neu starten)
 - Beim nächsten Neustart (nicht sofort)
5. Legen Sie die Neustart-Verteilungszeit fest. Sie können alle Maschinen zur gleichen Zeit neu starten oder die Updates zeitlich versetzt umsetzen.
6. Wählen Sie die Benachrichtigungsoptionen für die Benutzer:
 - Informieren der Benutzer über den Neustart der ihnen zugewiesenen Maschine
 - Vorlaufzeitraum für die Benachrichtigung
 - Benachrichtigungstext

Wenn eine Maschine eines Remote-PC-Zugriff-Maschinenkatalogs keinem Benutzer zugewiesen wurde, weist Studio vorübergehend die Maschine einer Remote-PC-Zugriff-Bereitstellungsgruppe zu, die dem Remote-PC-Zugriff-Maschinenkatalog zugeordnet ist. Dadurch erhält das System Informationen, die für die spätere Zuweisung an den geeigneten Benutzer benötigt werden. Neben der Bereitstellungsgruppenzuweisung wird die Zuordnung der Bereitstellungsgruppe zum Maschinenkatalog mit einem Prioritätswert verbunden. Je geringer der Wert, desto höher die Priorität. Dieser Prioritätswert kann nur mit dem PowerShell Software Developer Kit (SDK) wie unter [Informationen zum XenApp- und XenDesktop-SDK](#) beschrieben eingestellt werden.

Die Priorität bestimmt, welcher Bereitstellungsgruppe eine Maschine zugewiesen wird, wenn Folgendes eintritt:

- Die nicht konfigurierte Maschine wird beim System registriert
- Ein Benutzer benötigt eine Maschinenzuweisung

Wenn ein Remote-PC-Zugriff-Maschinenkatalog mehrere Bereitstellungsgruppenzuweisungen hat, wird die mit der höchsten Priorität vom System ausgewählt.

Sichern und Einschränken des Zugriffs auf Maschinen einer Bereitstellungsgruppe

Sep 29, 2015

Sie können alle ein- und ausgehenden Kommunikationsvorgänge mit Maschinen in einer Bereitstellungsgruppe mit dem SecureICA-Feature verbergen, mit dem das ICA-Protokoll verschlüsselt wird.

Wenn öffentliche Netzwerke beteiligt sind, empfiehlt Citrix, außer SecureICA noch zusätzliche Verschlüsselungsmethoden einzusetzen. Für Datenübertragungen über öffentliche Netzwerke empfiehlt Citrix die Verwendung der SSL/TLS-Verschlüsselung. Bei SecureICA wird die Datenintegrität auch nicht geprüft.

Standardmäßig ist SecureICA im System deaktiviert. Wenn Sie es aktivieren, ist der Standardverschlüsselungsgrad 128 Bit. Sie können den Verschlüsselungsgrad mit dem SDK konfigurieren. Weitere Informationen finden Sie unter [Informationen zum XenApp- und XenDesktop-SDK](#).

1. Wählen Sie in Studio den Knoten Bereitstellungsgruppen und dann die Bereitstellungsgruppe, deren Kommunikation Sie sichern möchten.
2. Klicken Sie auf Bereitstellungsgruppe bearbeiten und anschließend auf Grundeinstellungen.
3. Wählen Sie Secure ICA aktivieren.

Sie können den Zugriff auf Maschinen in einer Bereitstellungsgruppe einschränken. Unabhängig von der verwendeten Methode ersetzen jegliche vorgenommene Änderungen die vorherigen Einstellungen.

- Schränken Sie den Zugriff für Administratoren über Geltungsbereiche ein, um den Administratorzugriff auf Objektgruppen wie Maschinenkataloge, Bereitstellungsgruppen und Ressourcen zu steuern. Sie können einen Geltungsbereich erstellen und zuweisen, in dem Administratoren auf alle Anwendungen zugreifen können, und einen zweiten Geltungsbereich, der nur den Zugriff auf spezifische Anwendungen zulässt.
- Schränken Sie den Benutzerzugriff wie folgt ein:
 - SmartAccess-Richtlinienausdrücke, mit denen über NetScaler Gateway hergestellte Benutzerverbindungen gefiltert werden. Der Richtlinienadministrator kann diese Aufgabe im Knoten "Richtlinie" in Studio oder über Richtlinieneinstellungen durchführen, wie unter [Referenztabelle](#) beschrieben.
 - Ausschlussfilter für Zugriffsrichtlinien, die Sie mit dem Software Development Kit (SDK) festlegen. Zugriffsrichtlinien werden auf Bereitstellungsgruppen angewendet, um bestimmte Aspekte virtueller Desktopverbindungen genauer zu definieren. Sie können beispielsweise den Maschinenzugriff auf eine Untergruppe der Benutzer beschränken, die auf der Seite Endbenutzereinstellungen für die Bereitstellungsgruppe aufgeführt werden, sowie festlegen, welche Benutzergeräte eine Verbindung mit den Maschinen herstellen können. Mit Zugriffsrichtlinien können Sie ähnliche Ergebnisse erzielen; sie unterscheiden sich jedoch von Richtlinien.

Durch die Verwendung von Ausschlussfiltern können Zugriffsrichtlinien weiter angepasst werden. Aus Geschäfts- und Sicherheitsgründen können Sie den Zugriff für eine Untergruppe der Benutzer oder Geräte z. B. verweigern. Ausschlussfilter sind in der Standardeinstellung deaktiviert und können mit dem SDK festgelegt werden.

1. Wählen Sie in Studio den Knoten Bereitstellungsgruppen und dann die Bereitstellungsgruppe, deren Zugriff Sie einschränken möchten.

2. Klicken Sie auf Bereitstellungsgruppe bearbeiten und anschließend auf Geltungsbereiche.
3. Wählen Sie einen vorhandenen Geltungsbereich.
4. Fügen Sie die Objekte dem Geltungsbereich hinzu bzw. entfernen Sie sie.
5. Sie können Unterobjekte auswählen, indem Sie auf den Pfeil nach links klicken, die Unterobjekte auswählen und dann auf OK klicken.

SmartAccess-Richtliniausdrücke werden über NetScaler Gateway verwendet.

1. Wählen Sie in Studio unter Bereitstellungsgruppen die Gruppe aus, die Sie einschränken möchten.
2. Klicken Sie auf Bereitstellungsgruppe bearbeiten und anschließend auf Zugriffsrichtlinie.
3. Wählen Sie auf der Seite Zugriffsrichtlinie die Option Über NetScaler Gateway hergestellte Verbindungen aus. Es sind nur Verbindungen über NetScaler Gateway zulässig.
4. Wenn Sie nur einen Teil dieser Verbindungen auswählen möchten, wählen Sie Verbindungen, auf die mindestens einer der folgenden Filter zutrifft und tun Sie Folgendes:
 1. Definieren Sie die NetScaler Gateway-Site.
 2. Fügen Sie die SmartAccess-Richtliniausdrücke hinzu bzw. bearbeiten oder löschen Sie die Richtliniausdrücke, die für Benutzerzugriffsszenarios für die Bereitstellungsgruppe zulässig sind. Weitere Informationen zu NetScaler Gateway und SmartAccess-Richtliniausdrücken finden Sie unter [Configuring SmartAccess on NetScaler Gateway](#).

Ausschlussfilter werden mit dem SDK festgelegt.

Dieses Beispiel zeigt ein Unternehmensnetzwerk mit einem Lehlabor in einem Subnetz, von dem aus aber kein Zugriff auf eine bestimmte Bereitstellungsgruppe möglich sein soll, unabhängig davon, welcher Benutzer die Maschinen im Labor verwendet. Führen Sie dazu den folgenden SDK-Befehl aus:

```
Set-BrokerAccessPolicy -Name VPDesktops_Direct -ExcludedClientIPFilterEnabled $True -
```

Hinweis: Sie können das Sternchen (*) als Platzhalter verwenden, um alle Tags zu finden, die mit dem gleichen Richtliniausdruck beginnen. Wenn Sie beispielsweise auf einer Maschine das Tag VPDesktops_Direct hinzugefügt haben und auf einer anderen das Tag VPDesktops_Test, wird der Filter durch Festlegen des Tags auf VPDesktops_* im Skript Set-BrokerAccessPolicy auf beide Maschinen angewendet.

Weitere Informationen zum SDK finden Sie unter [Informationen zum XenApp- und XenDesktop-SDK](#).

Citrix empfiehlt, dass Sie diese Microsoft Richtlinie auf alle Benutzer anwenden, die auf Desktopbetriebssystemmaschinen zugreifen. Hiermit wird verhindert, dass Benutzer den Befehl Herunterfahren in einer Sitzung auswählen und den Desktop ausschalten, was ein manuelles Eingreifen des Systemadministrators erforderlich machen würde.

Navigieren Sie auf diese Richtlinie unter Benutzerkonfiguration\Administrative Vorlagen\Startmenü und Taskleiste\Befehl "Herunterfahren" entfernen und Zugriff darauf verweigern und setzen Sie den Befehl auf Aktiviert.

Herunterfahren und Neustart von Maschinen

Sep 29, 2015

Hinweis: Dieser Vorgang gilt nicht für Remote-PC-Zugriff-Maschinen.

1. Verwenden Sie in Studio Suchen, um die Maschine zu suchen, die Sie herunterfahren oder neu starten möchten, oder wählen Sie eine Bereitstellungsgruppe aus und klicken Sie auf Maschinen anzeigen.
2. Wählen Sie die Maschine aus und führen Sie eine der folgenden Aktionen aus. Abhängig vom Zustand der Maschine sind einige Optionen ggf. nicht verfügbar:
 - Herunterfahren erzwingen: Die Maschine wird zwingend abgeschaltet und die Liste der Maschinen wird aktualisiert.
 - Neu starten: Das Betriebssystem der Maschine wird heruntergefahren und sie wird dann neu gestartet. Wenn das Betriebssystem nicht heruntergefahren werden kann, bleibt die Maschine im aktuellen Zustand.
 - Anhalten: hält die Maschine an, ohne sie herunterzufahren und die Liste der Maschinen wird aktualisiert.
 - Herunterfahren: Das Betriebssystem der Maschine wird aufgefordert, herunterzufahren.

Hinweis: Wenn die Maschine nicht innerhalb von 10 Minuten heruntergefahren wird, wird sie ausgeschaltet. Wenn Windows versucht, während des Herunterfahrens Updates zu installieren, besteht die Gefahr, dass die Maschine ausgeschaltet wird, bevor die Updates abgeschlossen sind.

Mit diesem Feature konfigurieren Sie einen Neustartzeitplan für Serverbetriebssystemmaschinen in einer Bereitstellungsgruppe. Sie haben folgende Möglichkeiten:

- Geben Sie die Tage und die Stunde für den Neustart an
- Gleichmäßiges Verteilen des Neustarts über eine Zeitspanne (wichtig bei großen Gruppen)
- Erstellen einer benutzerdefinierten Meldung, die Benutzer über den Neustart informiert

Hinweis: Sie können kein automatisiertes Einschalten oder Herunterfahren in Studio durchführen.

1. Klicken Sie im Knoten Bereitstellungsgruppe auf Bereitstellungsgruppe bearbeiten.
2. Klicken Sie auf Neustartzeitplan.
3. Klicken Sie auf Ja. Die Optionen für den Neustartzeitplan werden aktiviert.
4. Geben Sie die folgenden Informationen ein:
 - Maschine neu starten: Wählen Sie Täglich oder geben Sie einen Wochentag an. Beispiel: Jeden Samstag.
 - Erste Gruppe neu starten um: Geben Sie im 24-Stundenformat die Stunde und Minute an, zu der die erste Gruppe neu gestartet wird. Beispiel: 20:15.
 - Weitere Gruppen neu starten alle: Wählen Sie "Alle Maschinen gleichzeitig neu starten" oder das Intervall, in dem Gruppen neu gestartet werden. Beispiel: 2 Stunden.
5. Legen Sie mit Neustartbenachrichtigung senden an Benutzer fest, wann eine Neustartbenachrichtigung an die Benutzer gesendet wird. Beispiel: Benutzer wird in 5 Minuten abgemeldet. Wenn Sie diese Option ausgewählt haben, geben Sie den Text für die Nachricht ein und klicken Sie auf OK.

Hinweis

Der Neustart wird anhand der auf den Maschinen in der Bereitstellungsgruppe festgelegten Zeitzone geplant

Aktualisieren einer Bereitstellungsgruppe

Sep 29, 2015

Maschinen in einer Bereitstellungsgruppe einer älteren Site enthalten möglicherweise ältere Virtual Desktop Agent-Versionen oder führen ältere Windows-Betriebssysteme aus. Dazu gehören:

- VDA-Version: 5.0 Service Pack 1, 5.5, 5.6 oder 5.6 Feature Pack 1
- VDA-Version 5.6 unter dem Betriebssystem Microsoft Windows 7
- Maschinen mit den Betriebssystemen Microsoft Windows XP oder Microsoft Windows Vista
Hinweis: Remote-PC-Zugriff wird unter Microsoft Windows Vista nicht unterstützt.

Führen Sie die folgenden Upgrades auf Maschinen aus, bevor Sie die Bereitstellungsgruppe aktualisieren:

- Aktualisieren Sie das Betriebssystem auf Microsoft Windows 7 oder Windows 8.x.
- Aktualisieren Sie den VDA auf Version 7 wie unter [XenDesktop 5-Upgradefaktoren](#) oder [Upgrade von XenDesktop 7](#) beschrieben.

Wenn Sie die Betriebssysteme oder die VDA-Versionen dieser Maschinen aktualisieren, aktualisieren Sie die Bereitstellungsgruppen, um die neuesten Funktionen dieses Releases zu nutzen. Beispielsweise wird durch das Upgrade einer Bereitstellungsgruppe die Möglichkeit aktiviert, StoreFront-Konfigurationen zuzuweisen.

Beachten Sie Folgendes zu Maschinenkatalog- und Bereitstellungsgruppen-Versionen:

- Maschinen, auf denen Windows XP, Windows Vista oder VDA-Version 5.6 oder frühere VDA-Versionen ausgeführt werden, können der XenDesktop 7.x- bzw. XenApp-Bereitstellungsgruppe nicht hinzugefügt werden. Beim Erstellen von Bereitstellungsgruppen können Sie die Option Einige der Maschinen führen Windows XP oder Windows Vista aus aktivieren, damit diese Maschinen in einer separaten Bereitstellungsgruppe sind.
- Eine Maschine mit VDA-Version 7.x kann zu jeder Bereitstellungsgruppe bzw. jedem Maschinenkatalog gehören.

Nach dem Upgrade des Betriebssystems der Maschine und der VDA-Version und vor dem Ausführen von Studio zum Upgrade der Bereitstellungsgruppe müssen Sie Folgendes durchführen:

- Stellen Sie sicher, dass Sie bei Verwenden von Provisioning Services für das Erstellen von Maschinenkatalogen und Bereitstellungsgruppen die VDA-Version über die Provisioning Services-Konsole aktualisieren.
- Starten Sie die Maschinen und registrieren sie beim Delivery Controller. Sonst kann der Studio-Assistent nicht ermitteln, ob die Maschinen in der Bereitstellungsgruppe aktualisiert werden müssen.

Stellen Sie sicher, dass die Maschinen, auf die von der Bereitstellungsgruppe verwiesen wird, wie folgt aktualisiert wurden:

- Microsoft Windows 7 oder Windows 8.x
 - Auf Maschinen mit Windows 7 muss ein Virtual Delivery Agent von XenDesktop 7.x oder XenApp 7.5 ausgeführt werden.
1. Klicken Sie im Knoten "Maschinen" auf Bereitstellungsgruppe.
 2. Wählen Sie die zu aktualisierende Bereitstellungsgruppe aus. Überprüfen Sie die Registerkarte "Details", auf der das Betriebssystem und die VDA-Version angezeigt wird, und klicken Sie auf Bereitstellungsgruppe aktualisieren. Der Assistent überprüft, ob die Bereitstellungsgruppe für ein Upgrade geeignet ist, und gibt an, dass die Bereitstellungsgruppe aktualisiert werden kann.
 - Wenn der Assistent angibt, dass die Gruppe für das Upgrade geeignet ist, wird eine Informationsmeldung angezeigt.

Klicken Sie auf Aktualisieren und folgen Sie den Anweisungen zum Abschluss des Upgrades.

- Wenn Maschinen nicht geeignet sind, führt der Assistent die Gründe für die Nichteignung der Maschine auf.
 - Klicken Sie für ausführliche Informationen auf Details anzeigen. Optional können Sie die Informationen in eine Datei exportieren, indem Sie im Dialogfeld Upgradewarnungen auf Details exportieren klicken.
 - Klicken Sie auf Abbrechen und aktualisieren Sie die entsprechenden Maschinen.

Citrix empfiehlt, dass Sie alle Maschinenprobleme beheben, bevor Sie die Bereitstellungsgruppen aktualisieren, damit alle Maschinen richtig funktionieren.

3. Konfigurieren Sie den StoreFront-Zugriff für die Bereitstellungsgruppe.
 1. Klicken Sie im Knoten Bereitstellungsgruppe auf eine Bereitstellungsgruppe und klicken Sie dann auf Bereitstellungsgruppe bearbeiten.
 2. Wählen Sie die StoreFront-URLs aus, die mit Push Citrix Receiver bereitgestellt werden, damit Receiver einen Store ohne Benutzereingriff verbinden kann. Sie können einen StoreFront-Server auf der Anzeige auswählen oder eine StoreFront-Serveradresse manuell eingeben.

Sie können die Maschinen in einer Bereitstellungsgruppe wieder in den vorherigen Zustand zurückversetzen.

1. Klicken Sie im Knoten "Bereitstellungsgruppe" auf Anzeigen.
2. Wählen Sie die zurückzusetzende Bereitstellungsgruppe aus. Überprüfen Sie die Registerkarte "Details", auf der das Betriebssystem und VDA-Version angezeigt wird, und klicken Sie auf Rückgängig machen. Folgen Sie den Anweisungen, um den Prozess abzuschließen.

Verwalten der Serverlast von Serverbetriebssystemmaschinen

Sep 29, 2015

Bei der Lastverwaltung wird die Serverlast gemessen und festgelegt, welcher Server unter den aktuellen Umgebungsbedingungen auszuwählen ist.

Wenn ein Benutzer sich an einer Serverbetriebssystemmaschine anmeldet, wird er von der Lastverwaltung dem Server zugewiesen, der der Anforderung am besten gerecht wird. Diese Auswahl basiert auf folgenden Faktoren:

- Status des Serverwartungsmodus (Standardeinstellung)
- Serverlastindex, d. h. eine Zahl, die die aggregierte Last angibt, basierend auf verschiedenen gemessenen Parametern, einschließlich CPU-Auslastung, Speichernutzung und Datenträgerverwendung. Lastindizes werden anhand einer Formel berechnet, dem Lastauswertungsprogramm, das durch Citrix Richtlinien festgelegt wird.
- Toleranzwert für gleichzeitige Anmeldungen, d. h. die zulässige Anzahl von gleichzeitigen Anmeldeanforderungen am Server. (Standardeinstellung)

Die Einstellungen für den Status des XenDesktop-Serverwartungsmodus und die Microsoft Windows-Remotedesktopverbindung beeinflussen, ob die Serverbetriebssystemmaschine bei der Lastverwaltung berücksichtigt wird.

Der Wartungsmodus ist aktiviert, wenn eine der folgenden Bedingungen gilt:

- XenDesktop-Serverwartungsmodus ist auf Ein gesetzt
- Remotedesktopverbindung ist auf Keine Verbindung mit diesem Computer zulassen gesetzt
- Remotedesktopverbindung ist nicht auf Keine Verbindung mit diesem Computer zulassen gesetzt und für den Anmeldemodus der Remotehostkonfiguration gilt eine der folgenden Optionen:
 - Wiederverbindungen zulassen, doch neue Anmeldungen verhindern
 - Wiederverbindungen zulassen, doch neue Anmeldungen bis zum Neustart des Servers verhindern

Die Serverbetriebssystemmaschine wird nur für den Lastausgleich berücksichtigt, wenn der Wartungsmodus deaktiviert ist.

Der Serverlastindex bestimmt, mit welcher Wahrscheinlichkeit ein Server, der Serverbetriebssystemmaschinen bereitstellt, Verbindungen erhält. Der Index setzt sich zusammen aus:

- Anzahl der Sitzungen
- Einstellungen für Leistungswerte wie CPU-, Datenträger- und Speichernutzung

Sie können diese Lastauswertungsprogramme über die Einstellungen der Lastverwaltungsrichtlinie konfigurieren, wie unter [Einstellungen der Richtlinie "Lastverwaltung"](#) beschrieben.

Sie können diesen Index wie folgt überwachen:

- Suche in Studio

Standardmäßig wird die Serverlastindexspalte ausgeblendet. Einschließen dieses Attributs in die Anzeige:

1. Wählen Sie eine Maschine aus.
2. Klicken Sie mit der rechten Maustaste auf eine Spaltenüberschrift und wählen Sie Spalte auswählen.
3. Führen Sie auf der Registerkarte für die ausgewählte Maschine den Bildlauf nach rechts durch, um für Serverlastindex das Maschinenattribut anzuzeigen.
4. Klicken Sie im Ordner "Maschine" auf Serverlastindex.

Sie können die Lastindexinformationen auch mit dem PowerShell SDK Get-BrokerMachine anzeigen. Weitere Informationen zum SDK finden Sie unter [Informationen zum XenApp- und XenDesktop-SDK](#).

Hinweis: Ein Serverlastindex von 10000 gibt Volllast für den Server an. Wenn keine anderen Server in der Site verfügbar sind, erhalten die Benutzer beim Starten einer Sitzung u. U. eine Meldung, dass der Desktop zurzeit nicht verfügbar ist.

- Director: Weitere Informationen zur Überwachung dieses Index finden Sie in der Dokumentation für [Director](#).

Sie steuern die Anzahl der ausstehenden Anmeldungen, die ein Server, der Serverbetriebssystemmaschinen bereitstellt, gleichzeitig akzeptieren kann, indem Sie den Toleranzwert für gleichzeitige Anmeldungen einstellen. Dadurch lassen sich Probleme mit der Serverleistung vermeiden.

Die Toleranzwerteinstellung für gleichzeitige Anmeldungen entspricht der Lastdrosselung in XenApp.

Wenn alle Server den Toleranzwert für gleichzeitige Anmeldungen erreichen oder überschreiten, wird die nächste Anmeldeanforderung dem Server mit der niedrigsten Anzahl ausstehender Anmeldungen zugewiesen. Wenn mehrere Server dieses Kriterium erfüllen, wird der Server mit dem niedrigsten Lastindex ausgewählt.

Die Richtlinien für Serverbetriebssystemmaschine legen fest, welche Einstellungen bei der Serverlastberechnung einbezogen werden.

Hinweis: Bei diesem Release wird die XenApp-Lastauswertungsprogrammeinstellung für XenApp 6.5 nicht unterstützt. Die standardmäßigen Lastverwaltungseinstellungen sind für die meisten Serverbetriebssystemmaschinen-Umgebungen geeignet. Wenn Sie diese Einstellungen ändern möchten, stehen bestimmte Richtlinien zur Verfügung. Weitere Informationen finden Sie unter [Einstellungen der Richtlinie "Lastverwaltung"](#).

Server für die Bereitstellung von Serverbetriebssystemmaschinen verfügen über Standardeinstellungen für die Lastverwaltung, die eine ausgewogene Verteilung der Sitzungen zwischen Servern gewährleistet. Mit der Lastverwaltungsrichtlinie können Sie die folgenden Einstellungen ändern.

Hinweis: Ein Wert von -1 für die folgenden Einstellungen gibt an, dass die jeweilige Einstellung bei der Lastberechnung nicht berücksichtigt wird.

Richtlinieneinstellung	Lastverwaltungsfaktor	Beschreibung	Wert
Toleranzwert für gleichzeitige Anmeldungen	Die Anzahl der gleichzeitigen Anmeldungen, die von einem Server akzeptiert werden können	Aktiviert (Standard): gibt die maximale Anzahl gleichzeitiger Anmeldungen vor Deaktiviert: Schließt diese Einstellung aus der Lastberechnung aus	Positive ganze Zahl (Standard = 2)
CPU-Nutzung	CPU-Nutzung in Prozent	Aktiviert: definiert den CPU-Nutzungswert (in Prozent), bei dem der Server Volllast	1 bis 100 Prozent

Richtlinieneinstellung	Lastverwaltungsfaktor	^{meldet} Beschreibung	(Standard = 90) Wert
		Deaktiviert (Standard): schließt die CPU-Nutzung aus der Lastberechnung aus	
CPU-Nutzung ausschließlich Prozesspriorität	Priorität, bei der die CPU-Nutzung für einen Prozess vom CPU-Nutzungslastindex ausgeschlossen wird	Aktiviert (Standard): Prozesse werden basierend auf dem ausgewählten Wert vom CPU-Nutzungslastindex ausgeschlossen Deaktiviert: Ignoriert die Konfiguration dieser Einstellung	Unter normal oder niedrig: schließt Prozesse aus, die eine Priorität von Unter normal oder "Niedrig" haben Niedrig: schließt Prozesse aus, die eine Priorität von Niedrig haben (Standard: Unter normal oder niedrig)
Datenträgernutzung	Datenträger-Warteschlangenlänge	Aktiviert: Definiert die Länge der Datenträgerwarteschlange, bei der ein Server 75 % Volllast meldet Deaktiviert (Standard): Schließt die Datenträgernutzung aus der Lastberechnung aus	Ganzzahliger Wert (Standard = 8)
Speichernutzung	Speichernutzung in Prozent	Aktiviert: bezieht die Speichernutzungsdaten in Lastberechnungen ein Deaktiviert (Standard): schließt die Speichernutzungsdaten in Lastberechnungen aus	1 bis 100 Prozent (Standard = 90)
Speichernutzung - Ausgangslast	Speicherauslastung in MB	Definiert die Speichernutzung in MB bis zu dem Punkt, an dem der Server aufgrund der Speichernutzung keine Last meldet. Grundlegende Betriebssystemfunktionen beanspruchen häufig mehrere 100 MB Arbeitsspeicher. Aktiviert (Standard): Schließt die Annäherung der Speichernutzung durch grundlegende Betriebssystemfunktionen	1 bis installierter Arbeitsspeicher in MB (Standard = 768)

Richtlinieneinstellung	Lastverwaltungsfaktor	vom Serverastindex aus Beschreibung	Wert
		Deaktiviert: Schließt den Wert für Speichernutzung - Ausgangslast nicht aus	
Sitzungshöchstanzahl	Maximale Anzahl von Sitzungen auf einem Server	Aktiviert (Standard): Gibt die maximale Anzahl gleichzeitiger Sitzungen auf dem Server vor Deaktiviert: Schließt diese Einstellung aus der Lastberechnung aus	1 bis maximale ganze Zahl (Standard = 250)

Ändern der Richtlinieneinstellungen für die Lastverwaltung

1. Rufen Sie in Studio den Assistenten für neue Richtlinien über den Knoten Richtlinie auf.
2. Wählen Sie den Knoten "Kategorien" und dann Lastverwaltung aus. Die in der obigen Tabelle beschriebenen Lastverwaltungseinstellungen werden angezeigt.
3. Klicken Sie auf Hinzufügen, um das Fenster Einstellungen bearbeiten aufzurufen.
4. Standardmäßig ist die Einstellung aktiviert. Sie können die Einstellung ändern.
5. Zum Wiederherstellen der Standardeinstellung klicken Sie auf Standardwert verwenden.

Delivery Controller-Umgebung

Sep 29, 2015

In einer Bereitstellung ist der Delivery Controller die serverseitige Komponente, die für die Verwaltung des Benutzerzugriffs sowie das Brokering und Optimieren von Verbindungen zuständig ist. Controller stellen auch die Maschinenerstellungsdienste zur Erstellung von Desktop- und Serverimages bereit.

Eine Site muss mindestens über einen Delivery Controller verfügen. Nach der Installation des ersten Controllers und der Erstellung einer Site können Sie weitere Controller hinzufügen. Es gibt zwei Hauptvorteile, mehr als einen Controller in einer Site zu haben.

- Redundanz: Als bewährte Methode sollte eine Produktionssite immer mindestens zwei Controller auf unterschiedlichen physischen Servern haben. Wenn ein Controller ausfällt, können die anderen die Verwaltung der Verbindungen und der Site übernehmen.
- Skalierbarkeit: Je intensiver die Aktivität einer Site, umso mehr nehmen CPU-Auslastung auf dem Controller und SQL Server-Datenbankaktivität zu. Zusätzliche Controller bieten die Möglichkeit, mehr Benutzer, Anwendungen und Desktopanforderungen zu verarbeiten und die Reaktionszeit insgesamt zu verbessern.

VDAs können erst verwendet werden, wenn sie bei einem Controller der Site registriert wurden (Herstellen der Kommunikation). Zur Suche eines Controllers überprüft der VDA die Controllerliste ListOfDDCs. Die Liste "ListOfDDCs" enthält einen oder mehrere DNS-Einträge oder IP-Adressen, die den VDA an die Controller der Site verweisen. Um einen Lastausgleich zu erzielen, verteilt der VDA die Verbindungen automatisch über alle Controller in der Liste.

Neben ListOfDDCs gibt die Liste ListOfSIDs an, über welche Maschinensicherheits-IDs (SIDs) der VDA als Controller angesprochen werden kann. Die Liste "ListOfSIDs" kann verwendet werden, um die Last auf Active Directory zu verringern oder um Sicherheitsbedrohungen durch einen nicht sicheren DNS-Server zu vermeiden.

Die Listen "ListOfDDCs" und "ListOfSIDs" müssen auf allen VDAs auf dem neuesten Stand sein, wenn Controller in der Site hinzugefügt und entfernt werden. Sind die Listen nicht aktuell, werden Sitzungsstarts, die durch einen nicht aufgeführten Controller vermittelt werden, vom VDA möglicherweise abgelehnt. Ungültige Einträge in der Liste können den Start der Systemsoftware des virtuellen Desktops verzögern. Halten Sie die Listen wie folgt auf dem neuesten Stand:

- Verwenden Sie automatische Updates zur automatischen Aktualisierung von ListOfDDCs und ListOfSIDs, wenn Controller hinzugefügt oder entfernt werden. Standardmäßig sind automatische Updates aktiviert.
- Eigenverwaltung: Aktualisieren Sie die Richtlinien- oder Registrierungseinstellungen für die Controller manuell.

Die Informationen in den Listen "ListOfDDCs" und "ListOfSIDs" können aus verschiedenen Orten der Bereitstellung stammen. Folgende Speicherorte werden vom VDA nacheinander überprüft, bis die Listen gefunden werden:

1. Ein Speicherort für den persistenten Speicher für automatische Updates. Dieser Speicherort enthält die Controllerinformationen, wenn automatische Updates aktiviert sind und nachdem der VDA erfolgreich zum ersten Mal nach der Installation registriert wurde. (Hier werden auch Maschinenrichtlinieninformationen gespeichert, sodass Richtlinieneinstellungen bei Neustarts beibehalten werden.)
Bei der ersten Registrierung nach der Installation oder bei deaktivierten automatischen Updates werden folgende Speicherorte vom VDA geprüft:
2. Richtlinieneinstellungen (Controller, Controller-SIDs).
3. Controllerinformationen des Virtual Desktop Agent-Schlüssels in der Registrierung. Diese Werte stammen ursprünglich vom VDA-Installationsprogramm, basierend auf den Controllerinformationen, die Sie bei der Installation des VDAs

eingeben.

4. Auf Organisationseinheiten basierende Controller-Discovery. Dies ist eine ältere Methode die zugunsten der Abwärtskompatibilität beibehalten wird.
5. Die durch die Maschinenerstellungsdienste erstellte Datei Personality.ini.

Wenn in ListOfDDCs mehrere Controller angegeben sind, erfolgt die Verbindung mit ihnen durch den VDA in einer zufälligen Reihenfolge. Die Liste "ListOfDDCs" kann auch Controllergruppen enthalten, die an den Klammern um zwei oder mehr Controllereinträge zu erkennen sind. Der VDA versucht, eine Verbindung mit jedem Controller in einer Gruppe herzustellen, bevor er weitere Einträge in der Liste "ListOfDDCs" versucht.

Für XenDesktop-Benutzer, die ein Upgrade einer Version vor 7.0 durchgeführt haben, ersetzt das Feature für automatische Updates die CNAME-Funktion der Vorversion. Sie können die CNAME-Funktion ggf. wieder aktivieren, jedoch funktioniert das DNS-Aliasing nur konsistent, wenn Sie das AutoUpdate-Feature und die CNAME-Funktion nicht zusammen verwenden. Weitere Informationen zum erneuten Aktivieren der CNAME-Funktion finden Sie unter [CTX137960](#).

Die Richtlinieneinstellung zur Aktivierung/Deaktivierung der automatischen Updates ist standardmäßig aktiviert.

Bei folgenden Bereitstellungstypen sind keine automatischen Updates möglich, stattdessen muss die Eigenverwaltung verwendet werden:

- Bereitstellungen mit Controllergruppen
- Bereitstellungen, die aus Sicherheitsgründen ListOfSIDs verwenden (Bei Bereitstellungen mit ListOfSIDs zur Verringerung der Active Directory-Last können automatische Updates verwendet werden)
- Bereitstellungen, die Provisioning Services ohne Datenträger für Zurückschreiben verwenden
- Bereitstellungen, die die Controller- oder Controller-SIDs-Richtlinieneinstellung verwenden

Die Richtlinieneinstellung Automatische Controllerupdates aktivieren ist Teil der Kategorie "Virtual Delivery Agent".

- Zur Verwendung automatischer Updates aktivieren Sie die Richtlinieneinstellung Automatische Controllerupdates aktivieren. Diese Einstellung ist standardmäßig aktiviert.
- Deaktivieren Sie die Richtlinieneinstellung Automatische Controllerupdates aktivieren, um automatische Updates zu deaktivieren.

Wird bei aktivierten automatischen Updates ein Virtual Delivery Agent (VDA) installiert, versucht der VDA, sich mit einem der Delivery Controller-Werte zu registrieren, die bei der Installation des VDAs von Ihnen eingegeben wurden. Das Installationsprogramm schreibt die Controllerinformationen, die Sie bei der Installation des VDAs angegeben haben, in den Registrierungswert "ListOfDDCs".

Nach der VDA-Registrierung sendet der Controller, bei dem die Registrierung erfolgte, eine Liste der vollständig qualifizierten Domännennamen (FQDNs) und Sicherheits-IDs (SIDs) an den VDA. Der VDA schreibt diese Liste in den persistenten Speicher für automatische Updates. Die Sitekonfigurationsdatenbank wird alle 90 Minuten von jedem Controller auf Controllerinformationen überprüft: Wenn seit der letzten Überprüfung ein Controller hinzugefügt oder entfernt wurde oder wenn eine Richtlinie geändert wurde, wird die aktualisierte Liste vom Controller an die registrierten VDAs gesendet. Der VDA nimmt alle Verbindungen von allen Controllern in der aktuellen Liste an.

Geht eine Liste ein, die den Controller, bei dem der VDA registriert ist, nicht enthält (d. h. der Controller wurde aus der Site entfernt), nimmt der VDA eine neue Registrierung bei einem der Controller aus der Liste vor. Nach der Registrierung bzw. Neuregistrierung erhält der VDA eine aktualisierte Liste.

Beispiel:

1. Eine Bereitstellung hat drei Controller: A, B und C. Ein VDA ist installiert und wird bei Controller B registriert (dies wurde bei der Installation des VDAs festgelegt).
2. Zwei Controller (D und E) werden der Site hinzugefügt. Innerhalb von 90 Minuten erhalten die VDAs aktualisierte Listen und können Verbindungen von Controllern A, B, C, D und E annehmen. (Die Last wird erst nach Neustart der VDAs gleichmäßig auf alle Controller verteilt.)
3. Controller B wird aus der Site entfernt. Innerhalb von 90 Minuten erhalten die VDAs aktualisierte Listen, da seit der letzten Überprüfung eine Controlleränderung stattfand. Der in Schritt 1 installierte VDA ist bei Controller B registriert, der jedoch in der Liste nicht mehr enthalten ist. Somit wird der VDA bei einem der anderen Controller der Liste (A, C, D und E) neu registriert.

Eigenverwaltung von Delivery Controllern

Wenn Sie keine automatischen Updates verwenden, müssen Sie die Citrix Richtlinieneinstellung oder die Registrierungswerte für jeden Virtual Delivery Agent (VDA) der Site (oder das VDA-Image) aktualisieren, wenn Delivery Controller in der Site hinzugefügt, verschoben oder entfernt wurden. Registrierungsänderungen können auch mit dem Gruppenrichtlinienobjekt aktualisiert werden.

Aktualisieren Sie die Werte der vollqualifizierten Domänennamen in der Richtlinieneinstellung Controller. Diese Richtlinieneinstellung ist in der Kategorie "Virtual Delivery Agent".

Wenn Sie in Ihrer Bereitstellung auch ListOfSIDs-Listen verwenden, aktualisieren Sie die SID-Werte der Richtlinieneinstellung Controller-SIDs.

Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab. Aktualisieren Sie den Registrierungsschlüssel "ListOfDDCs", der die vollqualifizierten Domänennamen aller Controller in der Site enthält. (Dieser Schlüssel entspricht der Active Directory-Site-Organisationseinheit.) Trennen Sie mehrere Werte mit Leerzeichen. Umschließen Sie Controllergruppen mit Klammern.

HKEY_LOCAL_MACHINE\Software\Citrix\VirtualDesktopAgent\ListOfDDCs (REG_SZ)

Wenn die Registrierungsstruktur HKEY_LOCAL_MACHINE\Software\Citrix\VirtualDesktopAgent sowohl die Registrierungsschlüssel "ListOfDDCs" und "FarmGUID" enthält, wird ListOfDDCs für Controller-Discovery verwendet; FarmGUID ist vorhanden, wenn die Organisationseinheit der Site bei der Installation des VDAs angegeben wurde.

Aktualisieren Sie wahlweise den Registrierungsschlüssel "ListOfSIDs":

HKEY_LOCAL_MACHINE\Software\Citrix\VirtualDesktopAgent\ListOfSIDs (REG_SZ)

Zum Hinzufügen, Entfernen oder Verschieben eines Delivery Controllers benötigen Sie die folgenden Rollen oder Berechtigungen:

Vorgang	Zweck	Serverrolle	Datenbankrolle
Datenbank erstellen	Geeignete, leere Datenbank erstellen	dbcreator	

Vorgang	Zweck	Serverrolle	Datenbankrolle
Schema erstellen	Erstellen aller dienstspezifischen Schemas und Hinzufügen des ersten Controllers zur Site	securityadmin *	db_owner
Controller hinzufügen	Hinzufügen eines Controller (ein anderer als der Erste) zur Site	securityadmin *	db_owner
Controller hinzufügen (Spiegelungsserver)	Hinzufügen einer Controller-Anmeldung zu dem Datenbankserver, der derzeit die gespiegelte Rolle einer gespiegelten Datenbank hat	securityadmin *	
Controller entfernen	Entfernen eines Controllers von der Site		db_owner
Schema aktualisieren	Anwenden von Aktualisierungen oder Hotfixes auf das Schema		db_owner
* Zwar ist die securityadmin-Serverrolle technisch restriktiver als die sysadmin-Serverrolle, aber in der Praxis ist sie als gleichwertig anzusehen. Wenn Sie mit Desktop Studio diese Vorgänge ausführen, muss das Benutzerkonto explizit Mitglied der sysadmin-Serverrolle sein.			

Vor dem Hinzufügen, Entfernen oder Verschieben von Controllern in einer Bereitstellung mit Datenbankspiegelung müssen Sie sicherstellen, dass sowohl die Haupt- als auch die Spiegeldatenbank ausgeführt wird. Wenn Sie mit Skripts für SQL Server Management Studio arbeiten, müssen Sie den SQLCMD-Modus vor dem Ausführen des Skripts aktivieren. Zum Prüfen der Spiegelung nach dem Hinzufügen, Entfernen oder Verschieben des Controllers führen Sie das PowerShell-Cmdlet `get-configdbconnection` aus, um sicherzustellen, dass der Failoverpartner in der Verbindungszeichenfolge für die Spiegelung eingerichtet wurde.

Gehen Sie nach dem Hinzufügen, Entfernen oder Verschieben eines Controllers wie folgt vor:

- Wenn das automatische Update aktiviert ist, erhalten die Virtual Delivery Agents (VDAs) eine aktualisierte Liste der Controller innerhalb von 90 Minuten.
- Ist das automatische Update nicht aktiviert, müssen Sie sicherstellen, dass die Controllerrichtlinieneinstellung oder der Registrierungsschlüssel "ListOfDDCs" für alle VDAs aktualisiert wird. Nachdem Sie einen Controller in eine andere Site verschoben haben, müssen Sie die Richtlinieneinstellung oder den Registrierungsschlüssel in beiden Sites aktualisieren.

Hinzufügen eines Controllers

Sie können keine Server, die mit einer früheren Version dieser Software installiert wurden, in einer Site hinzufügen, die mit dieser Version erstellt wurde.

1. Führen Sie auf dem Server, den Sie hinzufügen möchten, das Installationsprogramm aus und wählen Sie die Kernkomponenten aus, die Sie installieren möchten.
2. Klicken Sie in Studio auf Vorhandener Bereitstellung beitreten und geben Sie die Siteadresse ein.

Entfernen eines Controllers

Beim Entfernen eines Controllers werden die Citrix Software und andere Komponenten nicht deinstalliert. Es wird lediglich

der Controller aus der Datenbank entfernt, der damit für das Brokering von Verbindungen oder sonstige Aufgaben nicht mehr verfügbar ist. Wenn Sie einen Controller entfernen, können Sie diesen zu einem späteren Zeitpunkt der gleichen oder einer anderen Site wieder hinzufügen. Eine Site benötigt mindestens einen Controller. Aus diesem Grund können Sie den letzten in Studio aufgelisteten Controller nicht entfernen.

Hinweis: Stellen Sie sicher, dass der Controller eingeschaltet ist, sodass Studio in weniger als einer Stunde geladen wird. Wenn Studio den Controller lädt, den Sie entfernen möchten, schalten Sie den Controller aus, wenn Sie dazu aufgefordert werden.

Wenn Sie einen Controller von einer Site entfernen, wird die Controller-Anmeldung für den Datenbankserver nicht entfernt. Auf diese Weise wird vermieden, dass eine Anmeldung entfernt wird, die von den Diensten anderer Produkte auf demselben Computer verwendet wird. Die Anmeldung muss manuell entfernt werden, wenn sie nicht mehr erforderlich ist. Dazu benötigen Sie die Berechtigungen der securityadmin-Serverrolle.

Wichtig: Entfernen Sie den Controller erst dann aus Active Directory, wenn Sie ihn aus der Site entfernt haben.

1. Wählen Sie in Studio Konfiguration > Controller im linken Bereich aus und wählen Sie anschließend den Controller aus, den Sie entfernen möchten.
2. Klicken Sie im Aktionsbereich auf Controller entfernen. Wenn Sie nicht über die erforderlichen Datenbankrollen und Berechtigungen verfügen, können Sie ein Skript erstellen, mit dem der Datenbankadministrator den Controller für Sie entfernen kann.
3. Möglicherweise müssen Sie das Maschinenkonto des Controllers auf dem Datenbankserver entfernen. Bevor Sie dies durchführen, überprüfen Sie, ob das Konto von einem anderen Dienst verwendet wird.

Nachdem Sie mit Studio einen Controller entfernt haben, besteht ggf. kurze Zeit weiter Datenverkehr zu diesem Controller, um sicherzustellen, dass die aktuellen Tasks einwandfrei abgeschlossen werden. Wenn Sie das Entfernen eines Controllers in sehr kurzer Zeit erzwingen möchten, empfiehlt Citrix, den Server, auf dem er installiert war, herunterzufahren oder aus Active Directory zu entfernen. Starten Sie dann die anderen Controller in der Site neu, um sicherzustellen, dass keine weitere Kommunikation mit dem entfernten Controller stattfindet.

Verschieben eines Controllers in eine andere Site

Controller können nicht in eine Site verschoben werden, die mit einer früheren Version dieser Software erstellt wurde.

1. Wählen Sie in der derzeitigen Site des Controllers (der alten Site) in Studio Konfiguration > Controller im linken Bereich aus und wählen Sie anschließend den Controller aus, den Sie verschieben möchten.
2. Klicken Sie im Aktionsbereich auf Controller entfernen. Wenn Sie nicht über die erforderlichen Datenbankrollen und Berechtigungen verfügen, können Sie ein Skript erstellen, mit dem der Datenbankadministrator den Controller für Sie entfernen kann. Eine Site benötigt mindestens einen Controller. Aus diesem Grund können Sie den letzten in Studio aufgelisteten Controller nicht entfernen.
3. Öffnen Sie Studio auf dem zu verschiebenden Controller, setzen Sie bei entsprechender Aufforderung die Dienste zurück, klicken Sie auf Vorhandener Site beitreten und geben Sie die Adresse der neuen Site ein.

Sie können einen VDA zu einer anderen Site (von Site 1 zu Site 2) verschieben, wenn Sie ein Upgrade vornehmen oder wenn Sie ein in einer Testsite erstelltes VDA-Image in eine Produktionssite verschieben. Hierbei haben Sie zwei Möglichkeiten: das Installationsprogramm oder Citrix Richtlinien.

Verschieben eines VDAs mit dem Installationsprogramm

Wichtig: Geben Sie Delivery Controller im Installationsprogramm nur dann an, wenn die Richtlinieneinstellung "Controller" nicht verwendet wird.

Führen Sie das Installationsprogramm aus und fügen Sie einen Delivery Controller hinzu, wobei Sie in Site 2 einen vollqualifizierten Domännennamen (DNS-Eintrag) oder die IP-Adresse eines Controllers angeben.

Verschieben eines VDAs mit Citrix Richtlinien

Verschieben Sie mehrere VDAs mit dem Gruppenrichtlinien-Editor zwischen Sites.

Beispiel:

1. Erstellen Sie eine Richtlinie in Site 1 mit den nachfolgenden Einstellungen und filtern Sie die Richtlinie auf Bereitstellungsebene, um eine mehrstufige VDA-Migration zwischen den Sites zu erzielen.
 - Controller: mit vollqualifizierten Domännennamen (DNS-Einträgen) von einem oder mehreren Controllern der Site 2.
 - Automatische Controllerupdates aktivieren: auf "Deaktiviert" gesetzt.
2. Jeder VDA in der Bereitstellungsguppe wird innerhalb von 90 Minuten auf die neue Richtlinie hingewiesen. Der VDA ignoriert die eingegangene Liste der Controller (da automatische Updates deaktiviert sind) und wählt einen der in der Richtlinie angegebenen Controller, d. h. einen der Controller in Site 2.
3. Wenn der VDA erfolgreich bei einem Controller der Site 2 registriert wurde, empfängt er die Liste "ListOfDDCs" und die Richtlinieninformationen von Site 2, für die automatische Updates standardmäßig aktiviert sind. Da der Controller, bei dem der VDA in Site 1 registriert war, nicht in der vom Controller in Site 2 gesendeten Liste ist, erfolgt eine erneute Registrierung des VDAs unter Auswahl eines Controllers der Liste von Site 2. Ab sofort wird der VDA automatisch mit Informationen von Site 2 aktualisiert.

Diese Methode der Delivery Controller-Discovery wird vor allem unterstützt, um Abwärtskompatibilität zu gewährleisten und gilt nur für Virtual Delivery Agents (VDAs) für Windows-Desktops, aber nicht für VDAs für Windows-Server. Bei der Active Directory-basierten Discovery müssen alle Computer in einer Site Mitglied einer Domäne sein und zwischen der Domäne, die vom Controller verwendet wird, und den von den Desktops verwendeten Domänen muss eine Vertrauensstellung bestehen. Wenn Sie diese Methode verwenden, müssen Sie die GUID der Organisationseinheit in jeder Desktopregistrierung konfigurieren.

Führen Sie für eine auf Organisationseinheiten basierende Controller-Discovery das PowerShell-Skript `Set-ADControllerDiscovery.ps1` auf dem Controller aus (das Skript ist auf jedem Controller im Ordner `$Env:ProgramFiles\Citrix\Broker\Service\Setup Scripts`). Um das Skript auszuführen, müssen Sie über CreateChild-Berechtigungen für eine übergeordnete Organisationseinheit verfügen sowie über vollständige Administratorrechte.

Wenn Sie eine Site erstellen, müssen Sie eine entsprechende Organisationseinheit in Active Directory erstellen, wenn Desktops die Controller in der Site mit Active Directory ermitteln sollen. Die Organisationseinheit kann in jeder Domäne in der Gesamtstruktur erstellt werden, die Ihre Computer enthält. Optimalerweise sollte die Organisationseinheit auch die Controller der Site enthalten; dies wird jedoch nicht erzwungen bzw. ist nicht erforderlich. Ein Domänenadministrator mit den entsprechenden Privilegien kann die Organisationseinheit als leeren Container erstellen und dann Administrationsrechte an der Organisationseinheit an einen Citrix Administrator delegieren.

Das Skript erstellt einige wichtige Objekte. Nur Active Directory-Standardobjekte werden erstellt und verwendet. Es ist nicht erforderlich, das Schema zu erweitern.

- Eine Controller-Sicherheitsgruppe. Das Computerkonto aller Controller in der Site muss ein Mitglied dieser Sicherheitsgruppe sein. Desktops in einer Site nehmen nur Daten von Controllern an, die Mitglied dieser Sicherheitsgruppe sind.

Stellen Sie sicher, dass alle Controller auf allen virtuellen Desktops, auf denen der VDA ausgeführt wird, die Berechtigung "Auf diesen Computer vom Netzwerk aus zugreifen" haben. Geben Sie hierfür der Controller-Sicherheitsgruppe dieses Privileg. Wenn Controller nicht über diese Berechtigung verfügen, werden VDAs nicht registriert.

- Ein Dienstverbindungspunkt-Objekt, das Informationen zur Site enthält, wie den Namen der Site. Wenn Sie eine Organisationseinheit der Site mit dem Verwaltungstool "Active Directory-Benutzer und -Computer" prüfen, müssen Sie ggf. "Erweiterte Funktionen" im Menü "Ansicht" aktivieren, um Dienstverbindungspunkt-Objekte anzuzeigen.
- Einen "RegistrationServices" genannten Container, der in der Organisationseinheit der Site erstellt wird. Dieser Container enthält ein Dienstverbindungspunkt-Objekt für jeden Controller in der Site. Bei jedem Start des Controllers wird der Inhalt des Dienstverbindungspunkts geprüft und ggf. aktualisiert.

Wenn nach dem Abschluss der Erstinstallation mehrere Administratoren Controller hinzufügen oder entfernen, benötigen sie Berechtigungen zum Erstellen und Löschen von untergeordneten Objekten für den RegistrationServices-Container und zum Schreiben der Eigenschaften der Controller-Sicherheitsgruppe (diese Berechtigungen werden automatisch dem Administrator gewährt, der das Skript "Set-ADControllerDiscovery.ps1" ausführt). Der Domänenadministrator oder der Administrator der Originalinstallation kann diese Berechtigungen zuweisen; Citrix empfiehlt, dass Sie eine Sicherheitsgruppe für diesen Zweck einrichten.

Wenn Sie eine Organisationseinheit der Site verwenden:

- Informationen werden nur bei der Installation oder Deinstallation dieser Software in Active Directory geschrieben oder wenn beim Start eines Controllers die Informationen im Dienstverbindungspunkt aktualisiert werden müssen (der Controller wurde beispielsweise umbenannt oder der Kommunikationsport wurde geändert). In der Standardeinstellung erstellt das Skript "Set-ADControllerDiscovery.ps1" die richtigen Berechtigungen für die Organisationseinheit der Site und gibt jedem Controller Schreibrechte für den Dienstverbindungspunkt. Mit dem Inhalt der Objekte in der Organisationseinheit der Site wird eine Vertrauensstellung zwischen Desktops und Controllern erstellt. Stellen Sie Folgendes sicher:
 - Nur autorisierte Administratoren können der Controller-Sicherheitsgruppe über die Zugriffssteuerungsliste der Sicherheitsgruppe Computer hinzufügen oder Computer aus der Gruppe entfernen.
 - Nur autorisierte Administratoren und der relevante Controller können Informationen im Dienstverbindungspunkt des Controllers ändern.
- Bei einer Bereitstellung mit Replikation entstehen möglicherweise Verzögerungen; Weiteres hierzu finden Sie in der Microsoft Dokumentation. Dies ist besonders wichtig, wenn Sie die Organisationseinheit der Site in einer Domäne erstellen, die Domänencontroller in mehreren Active Directory-Sites enthält. Abhängig vom Standort der Desktops, der Controller und der Domänencontroller sind Änderungen, die Sie an Active Directory beim erstmaligen Erstellen der Organisationseinheit der Site, beim Installieren oder Deinstallieren von Controllern oder beim Ändern von Controllernamen oder Kommunikationsports vornehmen, erst nach der Replikation auf den entsprechenden Domänencontroller für Desktops erkennbar. Symptome einer solchen Replikationsverzögerung sind u. a. Desktops, die keine Verbindung mit Controllern herstellen können und daher nicht für Benutzerverbindungen verfügbar sind.
- Diese Software verwendet mehrere standardmäßige Computerobjektattribute in Active Directory zum Verwalten von Desktops. Je nach Bereitstellung kann der vollqualifizierte Domänenname des Maschinenobjekts so wie er im Active Directory-Datensatz des Desktops gespeichert ist in den Verbindungseinstellungen eingeschlossen werden, die an den Benutzer zum Herstellen einer Verbindung zurückgegeben werden. Stellen Sie sicher, dass diese Informationen mit den Informationen in der DNS-Umgebung übereinstimmen.

Verschieben eines Controllers mit auf Organisationseinheiten basierender Controller-Discovery in eine andere Site

Folgen Sie den Anweisungen unter [Verschieben eines Controllers](#). Nach dem Entfernen des Controllers aus der alten Site (Schritt 2), führen Sie folgendes PowerShell-Skript aus: `Set-ADControllerDiscover –sync`.

Das Skript synchronisiert die Organisationseinheit mit dem aktuellen Satz an Controllern. Nach dem Beitritt zur vorhandenen Site (Schritt 4), führen Sie das gleiche Skript auf einem Controller in der neuen Site aus.

Der XML-Dienst wird auf dem Delivery Controller ausgeführt und unterstützt die Protokolle HTTP und HTTPS.

Verwenden von SSL

Für HTTPS wird SSL (Secure Sockets Layer) vom XML-Dienst über Serverzertifikate, nicht aber über Clientzertifikate unterstützt. Zur Verwendung von HTTPS müssen Sie ein Serverzertifikat abrufen und es auf allen Controllern installieren und registrieren sowie einen Port mit dem SSL-Zertifikat konfigurieren.

- Wenn auf dem Controller IIS installiert ist, führen Sie die unter <http://support.microsoft.com/kb/299875> beschriebenen Schritte aus.
- Wenn der Controller nicht über IIS verfügt, können Sie das Zertifikat mit folgender Methode konfigurieren:
 1. Rufen Sie ein SSL-Serverzertifikat ab und installieren Sie es auf dem Controller, wie unter <http://blogs.technet.com/b/pki/archive/2009/08/05/how-to-create-a-web-server-ssl-certificate-manually.aspx> beschrieben. Weitere Informationen zum certreq-Tool finden Sie unter [http://technet.microsoft.com/en-us/library/cc736326\(Ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc736326(Ws.10).aspx).
 2. Konfigurieren Sie einen Port mit dem Zertifikat. Weitere Informationen finden Sie unter <http://msdn.microsoft.com/en-us/library/ms733791%28v=vs.110%29.aspx>.

Ändern der standardmäßigen HTTP- oder HTTPS-Ports

Der XML-Dienst lauscht standardmäßig an Port 80 auf HTTP-Datenverkehr und Port 443 auf HTTPS-Datenverkehr. Zwar können auch andere Ports auf einem Controller für den HTTP- oder HTTPS-Datenverkehr genutzt werden, dabei wird der Controller jedoch nicht vertrauenswürdigen Netzwerken ausgeliefert, und es entsteht ein Sicherheitsrisiko. Statt die Standardeinstellungen zu ändern, sollten Sie einen eigenständigen StoreFront-Server bereitstellen.

1. Führen Sie den folgenden Befehl auf dem Controller aus: `BrokerService.exe -WIPOrt http port -WISSLPORt https port`. `http port` ist die Portnummer für HTTP-Datenverkehr und `https port` ist die Portnummer für HTTPS-Datenverkehr.
2. Wenn der XML-Dienst den HTTP- oder HTTPS-Datenverkehr auf Standardports ignorieren soll, legen Sie die folgenden Registrierungswerte auf dem Controller fest und starten Sie den Broker-Dienst neu. Beide Werte sind in `HKLM\Software\Citrix\DesktopServer\`.

Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab.

 - Um den HTTP-Datenverkehr zu ignorieren, legen Sie für `XmlServicesEnableNonSsl` 0 fest.
 - Um den HTTPS-Datenverkehr zu ignorieren, legen Sie für `XmlServicesEnableSsl` 0 fest.

Drucker

Sep 29, 2015

Die Druckerverwaltung in Ihrer Umgebung umfasst verschiedene Stufen:

1. Planen der Druckarchitektur. Dazu gehört die Analyse folgender Faktoren: Unternehmensanforderungen, vorhandene Druckinfrastruktur, derzeitige Interaktion von Benutzern und Anwendungen mit Druckvorgängen und das für Ihre Umgebung am besten geeignete Druckverwaltungsmodell. Informationen zu Planungsstrategien finden Sie unter [Druckplanung](#).
2. Konfigurieren der Druckumgebung:
 - Wählen Sie eine Druckerbereitstellungsmethode aus (siehe [Bereitstellen von Druckern](#)).
 - Erstellen Sie die zur Bereitstellung des Druckkonzepts erforderlichen Richtlinien und aktualisieren Sie diese, wenn neue Benutzer oder Server hinzugefügt werden (siehe [Druckrichtlinien und Einstellungen](#)).
3. Testen einer Druckkonfiguration, bevor sie den Benutzern bereitgestellt wird.
4. Pflege der Citrix Druckumgebung:
 - [Verwalten von Druckertreibern](#)
 - [Optimieren der Druckleistung](#)
 - [Anzeigen von Druckern und Verwalten der Druckwarteschlangen](#)
5. Die Behandlung von Problemen, die in der Druckumgebung möglicherweise auftreten.

Bereitstellen von Druckern

Sep 29, 2015

Bei der Wahl der besten Drucklösung für Ihre Umgebung sollten Sie Folgendes berücksichtigen:

- Der universelle Druckserver bietet Features, die beim Windows-Druckanbieter nicht verfügbar sind: Zwischenspeichern von Bildern und Schriftarten, erweiterte Komprimierung, Optimierung und Unterstützung für QoS.
- Der universelle Druckertreiber unterstützt die von Microsoft definierten, öffentlichen geräteunabhängigen Einstellungen. Wenn Benutzer Zugriff auf die Geräteeinstellungen des Druckertreibers eines bestimmten Herstellers benötigen, stellt der universelle Druckserver gepaart mit einem Windows-systemeigenen Treiber die beste Lösung dar. In dieser Konfiguration bleiben die Vorteile des universellen Druckservers erhalten und die Benutzer können zugleich auf bestimmte Druckerfunktionen zugreifen. Allerdings ist zu bedenken, dass Windows-systemeigene Treiber wartungsbedürftig sind.

Um den universellen Druckserver mit einem Windows-systemeigenen Treiber zu verwenden, aktivieren Sie den universellen Druckserver. Wenn der Windows-systemeigene Treiber verfügbar ist, wird er standardmäßig verwendet. Andernfalls wird der universelle Druckertreiber verwendet. Um dieses Verhalten zu ändern, beispielsweise zur ausschließlichen Verwendung des Windows-systemeigenen Treibers oder des universellen Druckertreibers, müssen Sie die Richtlinieneinstellung Verwendung universeller Druckertreiber aktualisieren.

Konfigurieren des universellen Druckservers

Verwenden Sie die folgenden Citrix Richtlinieneinstellungen zum Konfigurieren des universellen Druckservers. Weitere Informationen finden Sie in der Onlinehilfe zu Richtlinieneinstellungen.

- Universellen Druckserver aktivieren: Der universelle Druckserver ist standardmäßig deaktiviert. Wenn Sie ihn aktivieren, müssen Sie festlegen, ob der Windows-Druckanbieter verwendet werden soll, wenn der universelle Druckserver nicht verfügbar ist. Nachdem der universelle Druckserver aktiviert wurde, können Benutzer Netzwerkdrucker über die Windows-Druckanbieter- und Citrix Anbieteroberflächen hinzufügen und auflisten.
- Port für Druckdatenstrom des universellen Druckservers (CGP): gibt die Nummer des TCP-Ports an, die vom Druckdatenstrom-Listener (CGP) des universellen Druckservers verwendet wird. Standardwert: 7229.
- Port für universellen Druckserverwebdienst (HTTP/SOAP): gibt die Nummer des TCP-Ports an, der vom Listener des universellen Druckservers für eingehende HTTP/SOAP-Anforderungen verwendet wird. Standardwert: 8080.
- Universeller Druckserver - Eingabebandbreitenlimit für Druckdatenstrom (KBit/s): gibt das obere Limit (in Kilobit pro Sekunde) für die Übertragungsrate der Druckdaten an, die von jedem Druckauftrag mit CGP an den universellen Druckserver übergeben werden. Standardwert: 0 (unbegrenzt).

Interaktion mit anderen Richtlinieneinstellungen

Der universelle Druckserver berücksichtigt andere Citrix Druckrichtlinieneinstellungen und interagiert mit ihnen wie in der folgenden Tabelle aufgeführt. Die Angaben basieren auf folgender Annahme: Die Richtlinieneinstellung "Universeller Druckserver" ist aktiviert, die Komponenten des universellen Druckservers sind installiert und die Richtlinieneinstellungen werden angewendet.

Richtlinieneinstellung	Interaktion

Client-druckerumleitung, Richtlinieneinstellung, automatisches Erstellen von Clientdruckern	Wenn der universelle Druckserver aktiviert ist, werden Clientnetzwerkdrucker mit dem universellen Druckertreiber statt den systemeigenen Treibern erstellt. Den Benutzern wird der gleiche Druckername wie zuvor angezeigt.
Sitzungsdrucker	Wenn Sie die Citrix Lösung des universellen Druckservers einsetzen, werden die Richtlinieneinstellungen für universelle Druckertreiber berücksichtigt.
Direkte Verbindungen zu Druckserver	Wenn der universelle Druckserver aktiviert ist und die Einstellung für die Richtlinie "Verwendung universeller Druckertreiber" für die ausschließliche Verwendung des universellen Drucks konfiguriert ist, kann mit dem universellen Druckertreiber ein direkter Netzwerkdrucker zum Druckserver erstellt werden.
UPD-Präferenz	Unterstützt EMF- und XPS-Treiber.

Auswirkungen auf Benutzeroberflächen

Der vom universellen Druckserver verwendete universelle Citrix Druckertreiber deaktiviert die folgenden Benutzeroberflächensteuerelemente:

- Schaltfläche für die lokalen Druckereinstellungen im Druckereigenschaften-Dialogfeld
- Schaltflächen für die lokalen Druckereinstellungen und die Vorschau im Dokumenteigenschaften-Dialogfeld

Beim universellen Druckserver gleicht der Assistent für die Druckerinstallation des Citrix Druckanbieters dem für den Windows-Druckanbieter mit den folgenden Ausnahmen:

- Beim Hinzufügen eines Druckers mit dem Namen oder einer Adresse können Sie eine HTTP/SOAP-Portnummer für den Druckserver angeben. Die Portnummer wird Teil des Druckernamens und wird angezeigt.
- Wenn in der Einstellung für die Richtlinie "Verwendung universeller Druckertreiber" festgelegt ist, dass universelles Drucken verwendet werden muss, wird der Name des universellen Druckertreibers bei der Auswahl des Druckers angezeigt. Der Windows-Druckanbieter kann den universellen Druckertreiber nicht verwenden.

Der Citrix Druckanbieter unterstützt kein clientseitiges Rendering.

Mit der automatischen Druckererstellung wird eine Liste der verfügbaren Drucker angelegt, die einem Benutzer nach der Anmeldung zur Verfügung stehen.

Konfigurieren des universellen Drucks

Verwenden Sie die folgenden Citrix Richtlinieneinstellungen zum Konfigurieren von des universellen Drucks. Weitere Informationen finden Sie in der Onlinehilfe zu Richtlinieneinstellungen.

- Verwenden universeller Druckertreiber: Mit dieser Einstellung legen Sie fest, wann das universelle Drucken verwendet wird.
- Automatisch generischen universellen Drucker erstellen: Mit dieser Einstellung aktivieren oder deaktivieren Sie die automatische Erstellung des generischen universellen Citrix Druckerobjekts für Sitzungen mit einem Benutzergerät, das mit Universal Printing kompatibel ist. Standardmäßig werden generische universelle Drucker nicht automatisch erstellt.
- Priorität universeller Treiber: Mit dieser Einstellung geben Sie an, in welcher Reihenfolge das System die universellen Druckertreiber verwendet, angefangen mit dem ersten Eintrag in der Liste. Sie können Treiber hinzufügen, bearbeiten oder entfernen und die Reihenfolge der Treiber in der Liste ändern.
- Universelles Drucken - Vorschau-einstellung Mit dieser Einstellung geben Sie an, ob die Druckvorschau für automatisch erstellte oder universelle Drucker verwendet werden soll.

- Universelles Drucken - EMF-Verarbeitungsmodus Mit dieser Einstellung steuern Sie die Verarbeitungsmethode für die EMF-Spooldatei auf dem Windows-Benutzergerät. Standardmäßig werden EMF-Datensätze direkt zum Drucker gespoolt. Direktes Spoolen an den Drucker ermöglicht eine schnellere Verarbeitung der Datensätze durch den Spooler und beansprucht weniger CPU-Ressourcen.
- Weitere Richtlinien finden Sie unter [Optimieren der Druckleistung](#).
- Informationen zum Ändern der Standardeinstellungen wie Papierformat, Druckqualität, Farbe, beidseitiger Druck und Anzahl der Exemplare finden Sie unter [How to Change the Default Settings on the Citrix Generic Universal Printer](#).

Erstellen von Druckern automatisch über das Benutzergerät

Zu Beginn einer Sitzung erstellt das System standardmäßig alle Drucker auf dem Benutzergerät automatisch. Sie können steuern, welche Typen der Drucker ggf. den Benutzern bereitgestellt werden und somit ein automatisches Erstellen verhindern.

Verwenden Sie die Citrix Richtlinieneinstellung Clientdrucker automatisch erstellen, um das automatische Erstellen zu steuern. Sie können Folgendes festlegen:

- Alle für das Benutzergerät sichtbaren Drucker, einschließlich der Netzwerkdrucker und der lokal angeschlossenen Drucker, werden zu Beginn einer Sitzung automatisch erstellt (Standardeinstellung)
- Alle lokalen Drucker, die physisch an das Benutzergerät angeschlossen sind, werden automatisch erstellt
- Nur der Standarddrucker für das Benutzergerät wird automatisch erstellt
- Automatische Erstellung ist für alle Clientdrucker deaktiviert

Die Einstellung Clientdrucker automatisch erstellen erfordert, dass für die Einstellung Clientdruckerumleitung Zugelassen (Standardeinstellung) festgelegt ist.

Konfigurieren der Umleitung bestimmter Drucker in Sitzungen

Wenn Sie vom Administrator zugewiesene Drucker erstellen möchten, müssen Sie die Citrix Richtlinieneinstellung Sitzungsdrucker konfigurieren. Verwenden Sie zum Hinzufügen eines Netzwerkdruckers zu dieser Richtlinie eine der folgenden Methoden:

- Geben Sie den UNC-Pfad im Format \\servername\printername ein.
- Navigieren Sie zu einem Drucker im Netzwerk.
- Navigieren Sie zu Druckern auf einem bestimmten Server. Geben Sie den Servernamen im Format \\servername an und klicken Sie auf Durchsuchen.

Wichtig: Der Server führt alle aktivierten Einstellungen für Sitzungsdrucker für alle angewendeten Richtlinien zusammen, angefangen von der höchsten bis zur niedrigsten Priorität. Ist ein Drucker in mehreren Richtlinienobjekten konfiguriert, werden angepasste Standardeinstellungen nur aus dem Richtlinienobjekt mit der höchsten Priorität verwendet, in dem der Drucker konfiguriert ist.

Welche Netzwerkdrucker über die Einstellung Sitzungsdrucker erstellt werden, kann je nachdem, wo die Sitzung gestartet wurde, durch Filtern, beispielsweise nach Subnetzen, variieren.

Angeben eines Standardnetzwerkdruckers für eine Sitzung

Standardmäßig wird der Hauptdrucker des Benutzers als Standarddrucker in der Sitzung verwendet. Verwenden Sie die Citrix Richtlinieneinstellung Standarddrucker, um die Auswahl des Standarddruckers auf dem Benutzergerät in einer Sitzung zu

ändern.

1. Wählen Sie unter Standarddrucker eine Einstellung für Standarddrucker des Clients wählen:
 - Netzwerkdruckername: Drucker, die mit der Richtlinieneinstellung Sitzungsdrucker hinzugefügt wurden, werden in diesem Menü angezeigt. Wählen Sie den als Standard für diese Richtlinie zu verwendenden Netzwerkdrucker aus.
 - Standarddrucker des Benutzers nicht anpassen: Verwendet die Einstellung der Terminaldienste oder des aktuellen Benutzerprofils für den Standarddrucker. Weitere Informationen finden Sie in der Onlinehilfe zu Richtlinieneinstellungen.
2. Wenden Sie die Richtlinie auf die Benutzergruppe (oder andere gefilterte Objekte) an, auf die sich auswirken soll.

Konfigurieren von Proximitydrucken

Hinweis: Proximitydrucken wird ebenfalls vom universellen Citrix Druckserver bereitgestellt; die hier beschriebene Konfiguration ist dazu nicht erforderlich.

1. Erstellen Sie eine separate Richtlinie für jedes Subnetz (oder entsprechend dem Druckerstandort).
2. Fügen Sie in jeder Richtlinie der Einstellung Sitzungsdrucker die Drucker an dem geografischen Standort des Subnetzes hinzu.
3. Setzen Sie die Einstellung Standarddrucker auf Standarddrucker des Benutzers nicht anpassen.
4. Filtern Sie die Richtlinien nach Client-IP-Adresse. Aktualisieren Sie diese Richtlinien, um Änderungen der DHCP-IP-Adressbereiche zu berücksichtigen.

Verwalten von Druckertreibern

Sep 29, 2015

Citrix empfiehlt die Verwendung des universellen Citrix Druckertreibers, um den Verwaltungsaufwand und mögliche Probleme mit Druckertreibern gering zu halten.

Wenn die automatische Erstellung fehlschlägt, installiert das System standardmäßig einen bei Windows integrierten systemeigenen Druckertreiber. Falls kein Treiber verfügbar ist, greift das System automatisch auf den universellen Druckertreiber zurück. Weitere Informationen über die Druckertreiberstandardeinstellungen finden Sie unter [Standarddruckvorgänge](#).

Wenn der universelle Druckertreiber von Citrix nicht für alle Szenarios geeignet ist, reduzieren Sie die Anzahl installierter Treiber auf Serverbetriebssystemmaschinen mit von Druckertreiberzuordnungen. Außerdem bietet die Zuordnung von Druckertreibern folgende Optionen:

- Beschränken bestimmter Drucker auf die ausschließliche Verwendung des universellen Citrix Druckertreibers
- Zulassen oder Verhindern der Erstellung von Druckern mit einem bestimmten Treiber
- Ersetzen veralteter oder beschädigter Treiber durch gewünschte Druckertreiber
- Ersetzen von Clienttreibernamen durch einen unter Windows Server verfügbaren Treiber

Die automatische Installation von Druckertreibern muss deaktiviert werden, damit Konsistenz zwischen Serverbetriebssystemmaschinen gewährleistet ist. Dies kann über Citrix Richtlinien und/oder Microsoft-Richtlinien erreicht werden. Zum Verhindern der automatischen Installation Windows-systemeigener Druckertreiber deaktivieren Sie die Citrix Richtlinieneinstellung Automatische Installation von mitgelieferten Druckertreibern.

Jeder Client liefert bei der Anmeldung Informationen zu den clientseitigen Druckern, einschließlich dem Namen des Druckers. Bei der automatischen Erstellung der Clientdrucker werden die Namen der Druckertreiber auf dem Windows-Server ausgewählt, die den Namen der Druckermodelle entsprechen, die der Client bereitgestellt hat. Beim automatischen Erstellen werden mit diesen identifizierten verfügbaren Druckertreibern umgeleitete Clientdruckwarteschlangen erstellt.

Nachfolgend wird allgemein beschrieben, wie Regeln für die Treiberersetzung definiert und Druckereinstellungen für zugeordnete Clientdruckertreiber bearbeitet werden.

1. Legen Sie die Regeln für die Treiberersetzung für automatisch erstellte Drucker fest, indem Sie die Citrix Richtlinieneinstellung Druckertreiberzuordnung und -kompatibilität konfigurieren. Fügen Sie dabei den Namen des Clientdruckertreibers hinzu und wählen Sie über das Menü Druckertreiber suchen den Servertreiber aus, durch den Sie den Clientdruckertreiber ersetzen möchten. Sie können in dieser Einstellung Platzhalter verwenden. Damit beispielsweise alle HP-Drucker einen bestimmten Treiber verwenden, geben Sie in der Richtlinieneinstellung HP* an.
2. Zum Ausschließen eines Druckertreibers wählen Sie den Treibernamen aus und aktivieren Sie die Einstellung Nicht verwenden.
3. Sie können bei Bedarf eine Treiberzuordnung bearbeiten, eine Zuordnung löschen oder die Reihenfolge der Treibereinträge in der Liste ändern.
4. Zum Bearbeiten der Druckereinstellungen für zugeordnete Clientdruckertreiber wählen Sie den Druckertreiber aus, klicken Sie auf Einstellungen und geben Sie die Einstellungen wie Druckqualität, Ausrichtung und Farbe an. Wenn Sie eine

Druckoption angeben, die der Druckertreiber nicht unterstützt, hat die Option keine Auswirkung. Mit dieser Einstellung werden die gespeicherten Druckereinstellungen überschrieben, die der Benutzer in einer vorherigen Sitzung festgelegt hat.

5. Citrix empfiehlt, das Verhalten der Drucker nach der Zuordnung von Treibern ausführlich zu testen, da einige Druckfunktionen möglicherweise nur über einen bestimmten Treiber zur Verfügung stehen.

Bei der Benutzeranmeldung wird die Clientdruckerkompatibilitätsliste vom System überprüft, bevor die Clientdrucker eingerichtet werden.

Optimieren der Druckleistung

Sep 29, 2015

Verwenden Sie den universellen Druckserver und den universellen Druckertreiber, um die Leistung zu optimieren. Die folgenden Richtlinien steuern die Druckoptimierung und Komprimierung:

- Universelles Drucken - Optimierungsstandards. Gibt die Standardeinstellungen für den universellen Drucker an, wenn er für eine Sitzung erstellt wird:
 - Mit Gewünschte Bildqualität geben Sie das standardmäßige Bildkomprimierungslimit an, das auf universelles Drucken angewendet wird. In der Standardeinstellung ist Standardqualität aktiviert, d. h. Benutzer können Bilder nur mit der Standardqualitäts- oder geringeren Qualitätskomprimierung drucken.
 - Mit Heavyweight-Komprimierung aktivieren oder deaktivieren Sie das Verringern der Bandbreite unter den Komprimierungsgrad, der von "Gewünschte Bildqualität" festgelegt ist; Bildqualität geht nicht verloren. Standardmäßig ist die Heavyweight-Komprimierung deaktiviert.
 - Mit den Einstellungen Zwischenspeichern von Bildern und Schriftarten legen Sie fest, ob Bilder und Schriftarten, die mehrmals im Druckdatenstrom vorhanden sind, zwischengespeichert werden. Sie stellen damit sicher, dass jedes eindeutige Bild oder jede Schriftart nur einmal zum Drucker gesendet wird. Standardmäßig werden eingebettete Bilder und Schriftarten zwischengespeichert.
 - Mit Nicht-Administratoren können diese Einstellungen ändern legen Sie fest, ob Benutzer die Standardeinstellungen für die Druckoptimierung in einer Sitzung ändern können. Standardmäßig können Benutzer die Standardeinstellungen für die Druckoptimierung nicht ändern.
- Universelles Drucken - Bildkomprimierungslimit. Definiert die maximale Qualität und die minimale Komprimierung für Bilder, die mit dem universellen Druckertreiber gedruckt werden. Das Limit für Bildkomprimierung ist standardmäßig auf Beste Qualität (verlustfreie Komprimierung) gesetzt.
- Universelles Drucken - Druckqualitätslimit. Der Höchstwert für Punkte pro Zoll (dpi) zum Erstellen von Ausdrucken in einer Sitzung. In der Standardeinstellung ist kein Limit angegeben.

Standardmäßig werden alle für Netzwerkdrucker bestimmten Druckaufträge von der Serverbetriebssystemmaschine über das Netzwerk direkt an den Druckserver weitergeleitet. Erwägen Sie, Druckaufträge über die ICA-Verbindung zu leiten, wenn das Netzwerk hohe Latenz oder beschränkte Bandbreite aufweist. Deaktivieren Sie hierzu die Citrix RichtlinienEinstellung Direkte Verbindungen zu Druckservern. Bei einer ICA-Verbindung werden die Daten komprimiert gesendet, es wird somit weniger Bandbreite bei der Übertragung der Daten über das WAN gebraucht.

Beim Drucken von Dateien von Serverbetriebssystemmaschinen auf Benutzerdruckern können bei anderen virtuellen Kanälen (z. B. Video) aufgrund des Wettbewerbs um die Bandbreite Leistungsverringerungen entstehen, insbesondere dann, wenn Benutzer über langsamere Netze auf Server zugreifen. Um dies zu verhindern, können Sie die für das Drucken verwendete Bandbreite beschränken. Indem Sie die Datenübertragungsrate für den Druck einschränken, stellen Sie im HDX-Datenstrom eine größere Bandbreite für die Übertragung von Video, Tastatureingaben und Mausdaten zur Verfügung. Wichtig: Das Druckerbandbreitenlimit wird immer eingehalten, auch wenn keine anderen Kanäle verwendet werden. Verwenden Sie die nachfolgenden Einstellungen der Citrix Richtlinie Bandbreite, um die Druckerbandbreitenlimits für die Sitzung zu beschränken. Führen Sie diese Aufgabe mit Studio aus, um die Limits für die Site festzulegen. Wenn Sie Limits für einzelne Server festlegen möchten, führen Sie diese Aufgabe über die Gruppenrichtlinien-Verwaltungskonsolle in Windows lokal auf jeder Serverbetriebssystemmaschine aus.

- Die Einstellung Bandbreitenlimit für Druckerumleitung dient zur Angabe der zum Drucken verfügbaren Bandbreite in

Kilobits pro Sekunde (KBit/s).

- Die Einstellung Bandbreitenlimit für Druckerumleitung (Prozent) begrenzt die zum Drucken verfügbare Bandbreite auf einen Prozentanteil der insgesamt verfügbaren Bandbreite.

Hinweis: Zur Verwendung der Einstellung Bandbreitenlimit für Druckerumleitung (Prozent) müssen Sie auch die Einstellung Bandbreitenlimit für Sitzung insgesamt aktivieren.

Wenn Sie Werte für beide Einstellungen eingeben, wird die strengste Einstellung (mit dem niedrigeren Wert) angewendet.

Zum Abrufen von Echtzeitinformationen zur Druckbandbreite verwenden Sie Citrix Director.

Anzeigen von Druckern und Verwalten der Druckwarteschlangen

Sep 29, 2015

In der folgenden Tabelle wird aufgeführt, wo Sie in Ihrer Umgebung Drucker anzeigen und die Druckwarteschlangen verwalten können.

	Druckmodell	UAC aktiviert?	Speicherort
Clientdrucker (an das Benutzergerät angeschlossene Drucker)	Clientdruckmodell	On	Druckverwaltung: Snap-in der Microsoft Management Console
		Off	Vor Windows 8: Systemsteuerung Windows 8: Snap-In Druckverwaltung
Netzwerkdrucker (Drucker auf einem Netzwerkdruckserver)	Netzwerkdruckmodell	On	Druckserver > Druckverwaltung Snap-In der Microsoft Management Console
		Off	Druckserver > Systemsteuerung
Netzwerkdrucker (Drucker auf einem Netzwerkdruckserver)	Clientdruckmodell	On	Druckserver > Druckverwaltung Snap-In der Microsoft Management Console
		Off	Vor Windows 8: Systemsteuerung Windows 8: Snap-In Druckverwaltung
Lokale Netzwerkserverdrucker (Drucker von einem	Netzwerkdruckmodell	On	Druckserver >

Netzwerkdruckserver, die einer Serverbetriebssystemmaschine hinzugefügt werden)	Druckmodell	UAC aktiviert? Off	Systemsteuerung Speicherort Druckserver > Systemsteuerung

Hinweis: Druckwarteschlangen für Netzwerkdrucker, die das Netzwerkdruckmodell verwenden, sind privat und können nicht über das System verwaltet werden.

Verwalten von Citrix Richtlinien

Sep 29, 2015

Durch das Konfigurieren einer Citrix Richtlinie steuern Sie den Benutzerzugriff oder die Sitzungsumgebung. Citrix Richtlinien sind die effizienteste Methode zum Steuern der Verbindungs-, Sicherheits- und Bandbreiteneinstellungen.

Sie erstellen Richtlinien für bestimmte Benutzergruppen, Geräte oder Verbindungstypen. Jede Richtlinie kann mehrere Einstellungen enthalten.

Sie arbeiten mit Richtlinien in Studio oder in der Gruppenrichtlinien-Verwaltungskonsolle in Windows. Welche Konsole oder welches Tool Sie verwenden, hängt davon ab, ob Sie in der Netzwerkumgebung Microsoft Active Directory verwenden und ob Sie die Berechtigungen haben, Gruppenrichtlinienobjekte (GPOs) zu verwalten.

Wenn Sie ein Citrix Administrator ohne Berechtigung zum Verwalten von Gruppenrichtlinien sind, verwenden Sie Studio, um Richtlinien für Ihre Site zu erstellen. Mit Studio erstellte Richtlinien werden in der Sitedatenbank gespeichert und Updates werden per Push auf den virtuellen Desktop übertragen, wenn der virtuelle Desktop beim Broker registriert wird oder ein Benutzer eine Verbindung mit dem virtuellen Desktop herstellt.

Wenn Sie in Ihrer Netzwerkumgebung Active Directory verwenden und die für die Verwaltung von Gruppenrichtlinien erforderlichen Berechtigungen haben, verwenden Sie den Gruppenrichtlinien-Editor, um Richtlinien für Ihre Site zu erstellen. Die Einstellungen, die Sie konfigurieren, beeinträchtigen die Gruppenrichtlinienobjekte, die Sie in der Gruppenrichtlinien-Verwaltungskonsolle angeben.

Wichtig: Zum Konfigurieren einiger Einstellungen müssen Sie den Gruppenrichtlinien-Editor verwenden:

- Richtlinieneinstellungen, die auf die Registrierung virtueller Desktops bei einem Controller bezogen sind
- Richtlinieneinstellungen für die Microsoft Application Virtualization-Server (App-V)

Gruppenrichtlinieneinstellungen (GPOs) werden in der folgenden Reihenfolge verarbeitet:

1. Lokale GPO
2. XenApp- bzw. XenDesktop-Site-GPO (in der Sitedatenbank gespeichert)
3. GPOs auf Siteebene
4. GPOs auf Domänenebene
5. Organisationseinheiten

Bei einem Konflikt können Richtlinieneinstellungen, die zuletzt verarbeitet werden, vorher verarbeitete überschreiben. Das heißt, dass Richtlinieneinstellungen die folgende Rangfolge haben:

1. Organisationseinheiten
2. GPOs auf Domänenebene
3. GPOs auf Siteebene
4. XenApp- bzw. XenDesktop-Site-GPO (in der Sitedatenbank gespeichert)
5. Lokale GPO

Beispiel: Ein Citrix Administrator erstellt eine Richtlinie (Richtlinie A) über Studio, mit der die Clientdateiumleitung für die Vertriebsmitarbeiter des Unternehmens aktiviert wird. Gleichzeitig erstellt ein anderer Administrator eine Richtlinie (Richtlinie

B) mit dem Gruppenrichtlinien-Editor, mit der die Clientdateiumleitung für die Vertriebsmitarbeiter deaktiviert wird. Wenn sich die Vertriebsmitarbeiter an den virtuellen Desktops anmelden, wird Richtlinie B angewendet und Richtlinie A ignoriert. Dies geschieht, da Richtlinie B auf der Domänenebene und Richtlinie A auf der Ebene der XenApp- bzw. XenDesktop-Site-GPOs verarbeitet wurde.

Beachten Sie jedoch, dass die Citrix Sitzungseinstellungen die gleichen Einstellungen in einer Active Directory-Richtlinie oder einer Remotedesktop-Sitzungshostkonfiguration überschreiben, wenn ein Benutzer eine ICA- oder Remotedesktopprotokoll (RDP)-Sitzung startet. Zu diesen Einstellungen gehören solche, die mit typischen RDP-Clientverbindungseinstellungen zusammenhängen, wie Desktophintergrund, Menüanimationen und das Anzeigeverhalten bei Drag & Drop.

Der Prozess für das Konfigurieren von Richtlinien ist:

1. Erstellen Sie die Richtlinie.
2. Konfigurieren Sie Richtlinieneinstellungen.
3. Weisen Sie die Richtlinie Benutzer- und Maschinenobjekten zu.
4. Weisen Sie der Richtlinie eine Priorität zu.
5. Prüfen Sie die effektive Richtlinie durch Ausführen des Gruppenrichtlinien-Modellierungsassistenten.

Navigieren durch die Citrix Richtlinien und Einstellungen

Sep 29, 2015

In Studio finden Sie Richtlinien und Vorlagen unter dem Knoten "Richtlinien".

Im Bereich der Richtlinien und Vorlagen sind die Richtlinieneinstellungen je nach Funktionalität bzw. Feature, für die bzw. das sie gelten, in Kategorien eingeteilt. Beispiel: Alle Richtlinieneinstellungen für die Profilverwaltung sind im Bereich "Profilverwaltung".

Richtlinieneinstellungen können auf Maschinen oder Benutzer angewendet werden. In Studio definieren Computereinstellungen (Richtlinieneinstellungen für Maschinen) das Verhalten von virtuellen Desktops und werden beim Start eines virtuellen Desktops angewendet. Beachten Sie, dass diese Einstellungen auch angewendet werden, wenn keine aktiven Benutzersitzungen auf dem virtuellen Desktop durchgeführt werden. Benutzerrichtlinieneinstellungen definieren die Benutzererfahrung bei Verbindungen mit virtuellen Desktops über ICA. Benutzerrichtlinien werden immer dann angewendet, wenn ein Benutzer eine Verbindung mit einem virtuellen Desktop über ICA herstellt oder erneut herstellt. Wenn ein Benutzer eine Verbindung mit einem virtuellen Desktop über RDP herstellt oder sich direkt an der Konsole anmeldet, werden Benutzerrichtlinien nicht angewendet. Beim Anwenden von Richtlinien in der vorhandenen Umgebung bestimmt das System, welche Einstellungen sich auf Maschinen beziehen und welche auf Benutzer und wendet diese entsprechend an.

Active Directory-Richtlinien und -Einstellungen sind in zwei Hauptkategorien unterteilt: Computerkonfiguration und Benutzerkonfiguration.

In Studio greifen Sie auf Richtlinien und Einstellungen zu, indem Sie den Knoten Richtlinie im linken Bereich der Konsole und dann die Registerkarte Richtlinien oder Vorlagen auswählen.

Im Gruppenrichtlinien-Editor greifen Sie auf Richtlinien und Einstellungen zu, indem Sie in der Struktur unter Computerkonfiguration oder Benutzerkonfiguration den Knoten Citrix Richtlinien auswählen.

Die Registerkarte "Richtlinien" In Studio enthält eine Liste aller Richtlinien, die erstellt wurden. Rechts neben der Liste werden die folgenden Registerkarten angezeigt:

- Übersicht enthält allgemeine Informationen zur ausgewählten Richtlinie einschließlich Namen, Priorität, und ob die Richtlinie derzeit aktiviert oder deaktiviert ist.
- Einstellungen enthält eine Liste aller für die ausgewählte Richtlinie konfigurierten Einstellungen.
- Zugewiesen zu enthält eine Liste aller Benutzer- und Maschinenobjekte, denen die ausgewählte Richtlinie momentan zugewiesen ist.

Die Registerkarte "Vorlagen" enthält Listen der von Citrix bereitgestellten Vorlagen und benutzerdefinierter Vorlagen. Rechts neben der Liste werden die folgenden Registerkarten angezeigt:

- Beschreibung enthält eine Beschreibung der ausgewählten Vorlage und bietet Informationen dazu, warum diese zum Erstellen einer Richtlinie in der vorhandenen Umgebung geeignet sein könnte.
- Einstellungen enthält eine Liste aller für die ausgewählte Vorlage konfigurierten Einstellungen.

Sie können die von Ihnen erstellten Vorlagen und Richtlinien auf Einstellungen durchsuchen. Alle Suchen zeigen gefundene

Elemente schon während der Eingabe an. Suchen nach einer Einstellung in einer Richtlinie oder Vorlage

1. Wählen Sie die Richtlinie oder Vorlage aus, die Sie durchsuchen möchten.
2. Wählen Sie Richtlinie bearbeiten oder Vorlage bearbeiten.
3. Geben Sie auf der Seite Einstellungen den Namen der Einstellung ein, die Sie suchen.

Sie können Suchen folgendermaßen einschränken:

- Wählen Sie eine bestimmte Produktversion aus der Dropdownliste, um nur die Einstellungen zu suchen, die sich auf diese Produktversion beziehen.
- Aktivieren Sie das Kontrollkästchen Nur ausgewählte anzeigen, um nur die Einstellungen zu suchen, die der ausgewählten Richtlinie hinzugefügt wurden.
- Wählen Sie eine Kategorie wie Automatische Wiederverbindung von Clients oder Bandbreite damit nur die Einstellungen in dieser Kategorie durchsucht werden.

Wählen Sie Alle Einstellungen, damit der ganze Einstellungskatalog durchsucht wird.

Verwalten von Citrix Richtlinienvorlagen

Sep 29, 2015

Mit Richtlinienvorlagen können Sie Citrix Richtlinien zur effektiven Verwaltung der Benutzer in Ihrer Umgebung konfigurieren. Vorlagen enthalten vorkonfigurierte Einstellungen, die die Leistung für bestimmte Umgebungen oder Netzwerkprobleme optimieren, z. B. wenn die Bandbreite gering ist oder Benutzer eine hochwertige Benutzererfahrung benötigen. Richtlinien können auf Server- und Desktopbetriebssysteme oder einzelne Benutzersitzungen angewendet werden. Sie können Vorlagen auf folgende Art verwenden:

- Als Quelle für das Erstellen anderer Richtlinien
- Als Tool für den Vergleich vorhandener Richtlinien
- Als Bereitstellungs- oder Empfangsmethode für Richtlinienkonfigurationen von Citrix Support oder vertrauenswürdigen Drittanbietern

Mit Richtlinienvorlagen können Sie die folgenden Aufgaben ausführen:

- Erstellen neuer Vorlagen mit vorhandenen Vorlagen oder Richtlinien
- Erstellen von neuen Richtlinien mit vorhandenen Vorlagen
- Importieren und Exportieren von Vorlagen

Richtlinienvorlagen werden auf der Registerkarte "Vorlagen" in Studio und im Gruppenrichtlinien-Editor angezeigt. In Studio werden Vorlagen in einer Einzelliste angezeigt. Im Gruppenrichtlinien-Editor werden Computervorlagen angezeigt, wenn Sie mit Computerrichtlinien arbeiten. Benutzervorlagen werden angezeigt, wenn Sie mit Benutzerrichtlinien arbeiten.

Für die ausgewählte Vorlage werden die folgenden Registerkarten mit Informationen unter der Vorlagenliste angezeigt:

- Beschreibung: enthält eine präzise Übersicht über den Zweck der ausgewählten Vorlage.
- Einstellungen enthält eine Liste aller konfigurierten Einstellungen, deren aktuelle Werte und die Standardwerte der ausgewählten Vorlage.

Vorlagen und delegierte Administration

Richtlinienvorlagen sind lokale Dateien, die auf dem Computer, auf dem Studio ausgeführt wird (und nicht in der Sitedatenbank) gespeichert sind. Lokale Dateien werden nicht über die Rollen und Geltungsbereiche der delegierten Administration, sondern über Windows-Administratorrechte gesteuert. Dies bedeutet, dass ein Administrator, der über die delegierte Administration Leseberechtigungen erhalten hat, beispielsweise neue Vorlagen erstellen kann. Da Vorlagen jedoch lokale Dateien sind, werden an der Umgebung keine Änderungen vorgenommen.

Integrierte Vorlagen

Citrix bietet die folgenden integrierten Vorlagen:

- High Definition-Benutzererfahrung: Solche Vorlagen enthalten Einstellungen, die für Audio, Grafiken und Video eine hohe Qualität gewährleisten.
- Hohe Serverskalierbarkeit: Diese Vorlagen enthalten Einstellungen für eine optimale Benutzererfahrung in Szenarios, bei denen mehrere Benutzer auf einem Server gehostet werden.
- Optimierte Bandbreite für WAN: Diese Vorlagen enthalten Einstellungen für eine optimierte Benutzererfahrung bei geringer Bandbreite oder Verbindungen mit hoher Latenz, z. B. für Benutzer, die in einer Geschäftsstelle über eine gemeinsame WAN-Verbindung arbeiten.
- Sicherheit und Steuerung: Diese Vorlagen enthalten Einstellungen zum Deaktivieren des Zugriffs auf Peripheriegeräte, Laufwerkzuordnung, Portumleitung und Flash-Beschleunigung auf Benutzergeräten.

Sie können diese Vorlagen als Modell zum Erstellen neuer Richtlinien oder Vorlagen verwenden. Integrierte Vorlagen werden von Citrix erstellt und aktualisiert. Diese Vorlagen dürfen nicht geändert oder gelöscht werden. Sie können jedoch Vorlagen bearbeiten oder löschen, die Sie über Studio oder den Gruppenrichtlinien-Editor erstellen bzw. importieren.

Vorlagen werden von einer vorhandenen Vorlage oder Richtlinie erstellt. Die neue Vorlage enthält dann dieselben Einstellungen wie die Originalvorlage oder -richtlinie. Alle Zuordnungen in der ursprünglichen Richtlinie werden nicht in die Vorlage eingeschlossen.

Erstellen einer neuen Vorlage basierend auf einer vorhandenen Vorlage

1. Wählen Sie in Studio im linken Bereich den Knoten Richtlinie aus.
2. Klicken Sie auf die Registerkarte Vorlagen und wählen Sie dann die Vorlage, von der Sie eine neue Vorlage erstellen möchten.
3. Klicken Sie auf Neue Vorlage. Der Assistent zum Erstellen einer neuen Vorlage wird angezeigt.
4. Wählen und konfigurieren Sie die Richtlinieneinstellungen, die Sie in der Vorlage einschließen möchten. Entfernen Sie vorhandene Einstellungen, die nicht eingeschlossen werden sollen.
5. Geben Sie einen Namen für die Vorlage ein.
6. Klicken Sie auf Fertig stellen. Die neue Vorlage wird auf der Registerkarte Vorlagen angezeigt.

Erstellen einer neuen Vorlage auf der Basis einer vorhandenen Richtlinie

1. Wählen Sie in Studio im linken Bereich den Knoten Richtlinie aus.
2. Wählen Sie auf der Registerkarte Richtlinien die Richtlinie aus, von der Sie die Vorlage erstellen möchten.
3. Klicken Sie auf Als Vorlage speichern. Das Dialogfeld Als Vorlage speichern wird angezeigt.
4. Wählen und konfigurieren Sie die Richtlinieneinstellungen, die Sie in der Vorlage einschließen möchten. Entfernen Sie vorhandene Einstellungen, die nicht eingeschlossen werden sollen.
5. Geben Sie einen Namen und eine Beschreibung für die neue Vorlage ein.
6. Klicken Sie auf Fertig stellen. Die neue Vorlage wird auf der Registerkarte Vorlagen angezeigt.

Richtlinienvorlagen stehen lokal für den Computer zur Verfügung, auf dem Sie die Konsole zum Verwalten der Site ausführen. Sie können Richtlinienkonfigurationen zwischen Umgebungen übertragen, u. a. zu anderen Sites, die Sie auf dem Computer verwalten, auf dem die Konsole ausgeführt wird.

Der Transfer von Vorlagen erfolgt über einen Import oder Export. Dies ermöglicht:

- Implementieren von Richtlinienkonfigurationen von Servern in anderen Sites
- Erstellen von Backups der Vorlagendateien zum Wiederherstellen von Richtlinienkonfigurationen
- Weitergeben der Richtlinienkonfigurationen der Site an den Citrix Support bei der Problembehandlung
- Implementieren von Richtlinienkonfigurationen, die vom Citrix Support zur Behebung von Problemen in der Site erstellt wurden

Importieren einer Vorlage

1. Wählen Sie in Studio im linken Bereich den Knoten Richtlinie aus.
2. Klicken Sie auf die Registerkarte Vorlagen und dann auf Vorlage importieren. Das Dialogfeld Vorlage importieren wird angezeigt.

3. Wählen Sie die Vorlagendatei, die Sie importieren möchten, und klicken Sie auf Öffnen. Die importierte Vorlage wird in der Liste "Vorlagen" angezeigt.

Hinweis: Wenn Sie eine Vorlage importieren, die denselben Namen wie eine vorhandene hat, können Sie die vorhandene Vorlage überschreiben oder die Vorlage unter einem anderen Namen speichern, der automatisch generiert wird.

Exportieren einer Vorlage

1. Wählen Sie in Studio im linken Bereich den Knoten Richtlinie aus.
2. Klicken Sie auf die Registerkarte Vorlagen und wählen Sie dann die Vorlage, die Sie exportieren möchten.
3. Klicken Sie auf Vorlage exportieren. Das Dialogfeld Vorlage exportieren wird angezeigt.
4. Legen Sie den Ordner fest, in dem die Vorlage gespeichert wird, und klicken Sie auf Speichern.

Im angegebenen Ordner wird eine GPT-Datei erstellt.

Erstellen von Richtlinien

Sep 29, 2015

Bevor Sie eine Richtlinie erstellen, entscheiden Sie, für welche Benutzergruppen oder Geräte sie gilt. Sie können Richtlinien basierend auf Aufgabenbereich, Verbindungstyp, Benutzergerät oder geografischer Position erstellen. Alternativ können Sie die gleichen Kriterien verwenden wie für Windows Active Directory-Gruppenrichtlinien.

Wenn Sie bereits eine Richtlinie für eine Gruppe erstellt haben, sollten Sie möglichst die Richtlinie bearbeiten und die entsprechenden Einstellungen konfigurieren, statt eine andere Richtlinie zu erstellen. Vermeiden Sie es, eine neue Richtlinie zu erstellen, deren einziger Zweck ist, eine bestimmte Einstellung zu aktivieren oder bestimmte Benutzer von der Richtlinie auszunehmen.

Sie können Richtlinien mit den folgenden Methoden erstellen:

- Erstellen einer neuen Richtlinie basierend auf den in einer Richtlinienvorlage enthaltenen Einstellungen
- Erstellen einer neuen Richtlinie mit dem Assistenten

Die neue Richtlinie enthält standardmäßig die gleichen Einstellungen wie die Originalvorlage. Sie können diese Einstellungen akzeptieren oder die Richtlinie Ihren Anforderungen anpassen.

1. Wählen Sie in Studio im linken Bereich den Knoten Richtlinie aus.
2. Klicken Sie auf die Registerkarte Vorlagen und wählen Sie dann die Vorlage, von der Sie die Richtlinie erstellen möchten.
3. Klicken Sie auf Richtlinie erstellen. Der Assistent zum Erstellen einer neuen Richtlinie wird angezeigt.
4. Wählen Sie, ob Sie die Richtlinie anpassen möchten.
 - Wenn Sie die Richtlinie nicht anpassen möchten, fahren Sie mit Schritt 6 fort.
 - Wenn Sie die Richtlinie anpassen möchten, klicken Sie auf Standardeinstellungen ändern und Einstellungen hinzufügen und fügen Sie dann nach Bedarf Einstellungen hinzu oder entfernen Sie diese.
5. Wählen Sie aus, wie die Richtlinie angewendet wird. Führen Sie einen der folgenden Schritte aus:
 - Klicken Sie auf Ausgewählten Benutzer- und Maschinenobjekten zuweisen und wählen Sie die Benutzer- und Maschinenobjekte, auf die Sie die Richtlinie anwenden möchten. Damit wird die Richtlinie nur auf die hier ausgewählten Benutzer- und Maschinenobjekte angewendet.
 - Klicken Sie auf Allen Objekten in der Site zuweisen. Damit wird die Richtlinie auf alle Benutzer- und Maschinenobjekte in der Site angewendet.
6. Geben Sie einen eindeutigen Namen für die neue Richtlinie ein oder akzeptieren Sie den automatisch generierten Standardnamen. Geben Sie optional eine Beschreibung ein.
Hinweis: Empfehlenswert sind Richtlinienennamen, die beschreiben, wer von der Richtlinie betroffen ist, z. B. Buchhaltung oder Remotebenutzer.
7. Lassen Sie die Richtlinie aktiviert oder deaktivieren Sie das Kontrollkästchen Richtlinie aktivieren, um die Richtlinie zu deaktivieren. Wenn die Richtlinie aktiviert ist, kann sie sofort auf Benutzer angewendet werden, die sich bei der Site anmelden. Deaktivieren der Richtlinie verhindert, dass sie angewendet wird. Wenn Sie die Priorität der Richtlinie ändern müssen oder später weitere Einstellungen hinzufügen möchten, können Sie die Richtlinie deaktivieren, bis Sie damit fertig sind, und die Richtlinie dann anwenden.

Mit dem Assistenten für neue Richtlinien können Sie eine neue, leere Richtlinie erstellen, der Sie die gewünschten Einstellungen hinzufügen.

1. Wählen Sie in Studio im linken Bereich den Knoten Richtlinie aus.
2. Klicken Sie auf Richtlinien manuell erstellen. Der Assistent zum Erstellen einer neuen Richtlinie wird angezeigt.
3. Fügen Sie individuelle Richtlinieneinstellungen nach Bedarf hinzu und konfigurieren Sie diese.
4. Wählen Sie aus, wie die Richtlinie angewendet wird. Führen Sie einen der folgenden Schritte aus:
 - Klicken Sie auf Ausgewählten Benutzer- und Maschinenobjekten zuweisen und wählen Sie die Benutzer- und Maschinenobjekte, auf die Sie die Richtlinie anwenden möchten. Damit wird die Richtlinie nur auf die hier ausgewählten Benutzer- und Maschinenobjekte angewendet.
 - Klicken Sie auf Allen Objekten in der Site zuweisen. Damit wird die Richtlinie auf alle Benutzer- und Maschinenobjekte in der Site angewendet.
5. Geben Sie einen eindeutigen Namen für die neue Richtlinie ein oder akzeptieren Sie den automatisch generierten Standardnamen. Geben Sie optional eine Beschreibung ein.

Hinweis: Empfehlenswert sind Richtliniennamen, die beschreiben, wer von der Richtlinie betroffen ist, z. B. Buchhaltung oder Remotebenutzer.
6. Lassen Sie die Richtlinie aktiviert oder deaktivieren Sie das Kontrollkästchen Richtlinie aktivieren, um die Richtlinie zu deaktivieren. Wenn die Richtlinie aktiviert ist, kann sie sofort auf Benutzer angewendet werden, die sich bei der Site anmelden. Deaktivieren der Richtlinie verhindert, dass sie angewendet wird. Wenn Sie die Priorität der Richtlinie ändern müssen oder später weitere Einstellungen hinzufügen möchten, können Sie die Richtlinie deaktivieren, bis Sie damit fertig sind, und die Richtlinie dann anwenden.

Konfigurieren von Richtlinieneinstellungen

Sep 29, 2015

Richtlinien enthalten Einstellungen, die auf Verbindungen angewendet werden, wenn die Richtlinie durchgesetzt wird. Richtlinieneinstellungen können deaktiviert, aktiviert oder nicht konfiguriert sein. Standardmäßig sind Richtlinieneinstellungen nicht konfiguriert, d. h. sie wurden keiner Richtlinie hinzugefügt. Einstellungen werden nur angewendet, wenn sie einer Richtlinie hinzugefügt wurden.

Manche Richtlinieneinstellungen können einen der folgenden Zustände haben:

- Mit Zugelassen oder Nicht zugelassen wird die durch die Einstellung gesteuerte Aktion ermöglicht oder verhindert.
- Mit Aktiviert oder Deaktiviert schalten Sie die Einstellung ein oder aus. Wenn Sie eine Einstellung deaktivieren, wird sie nicht durch Richtlinien mit geringerer Priorität aktiviert.

Bei Einstellungen, die zugelassen oder nicht zugelassen sind, wird die durch die Einstellung gesteuerte Aktion entweder erlaubt oder verhindert. In manchen Fällen dürfen Benutzer die Aktion der Einstellung in der Sitzung verwalten, in anderen dürfen sie das nicht. Wenn beispielsweise für Menüanimation die Option Zugelassen eingestellt ist, können Benutzer Menüanimationen in ihrer Clientumgebung steuern.

Manche Einstellungen steuern außerdem die Wirksamkeit von abhängigen Einstellungen. Die Einstellung Clientlaufwerkumleitung steuert beispielsweise, ob Benutzer auf die Laufwerke ihres Geräts zugreifen können. Damit Benutzer auf Netzlaufwerke zugreifen können, muss sowohl diese Einstellung als auch die Einstellung Clientnetzlaufwerke der Richtlinie hinzugefügt werden. Wenn die Einstellung Clientlaufwerkumleitung deaktiviert ist, können Benutzer nicht auf ihre Netzlaufwerke zugreifen, selbst wenn die Einstellung Clientnetzlaufwerke aktiviert ist.

In der Regel treten Änderungen an Richtlinieneinstellungen, die sich auf Maschinen auswirken, in Kraft, wenn der virtuelle Desktop neu gestartet wird oder wenn sich ein Benutzer anmeldet. Änderungen an Richtlinieneinstellungen, die Auswirkungen auf Benutzer haben, treten in Kraft, wenn sich die betroffenen Benutzer das nächste Mal anmelden. Wenn Sie Active Directory verwenden, werden die Richtlinieneinstellungen aktualisiert, wenn Active Directory die Richtlinien in regelmäßigen 90-Minuten-Intervallen erneut evaluiert. Die Richtlinieneinstellungen werden angewendet, wenn der virtuelle Desktop neu gestartet wird oder wenn sich ein Benutzer anmeldet.

Für manche Richtlinieneinstellungen können Sie einen Wert eingeben oder aus einer Liste wählen, wenn Sie die Einstellung der Richtlinie hinzufügen. Sie können die Konfiguration der Einstellung einschränken, indem Sie Standardwert verwenden wählen. Mit dieser Option deaktivieren Sie die Konfiguration der Einstellung und nur der Standardwert der Einstellung darf verwendet werden, wenn die Richtlinie angewendet wird. Die ist unabhängig von dem Wert, der vor dem Aktivieren von Standardwert verwenden eingegeben wurde.

Die Standardwerte für alle Citrix Richtlinieneinstellungen finden Sie in der [Referenz für Richtlinieneinstellungen](#).

Citrix empfiehlt beim Konfigurieren von Richtlinieneinstellungen Folgendes:

- Weisen Sie Richtlinien Gruppen statt einzelnen Benutzern zu. Wenn Sie Richtlinien Gruppen zuweisen, werden Zuweisungen automatisch aktualisiert, wenn Sie Benutzer Gruppen hinzufügen oder sie daraus entfernen.
- Aktivieren Sie nicht widersprechende oder überlappende Einstellungen in der Konfiguration des Remotedesktop-Sitzungshosts. In manchen Fällen bietet die Remotedesktop-Sitzungshostkonfiguration ähnliche Funktionalität wie Citrix Richtlinieneinstellungen. Wählen Sie nach Möglichkeit für alle Einstellungen den gleichen Status (aktiviert oder

deaktiviert), um die Problembehandlung zu erleichtern.

- Deaktivieren Sie Richtlinien, die nicht verwendet werden. Richtlinien, denen keine Einstellungen hinzugefügt wurden, verursachen unnötigen Verarbeitungsaufwand.

Vergleichen von Richtlinien und Vorlagen

Sep 29, 2015

In einigen Fällen müssen Sie die Einstellungen in einer Richtlinie oder Vorlage mit denen in anderen Richtlinien oder Vorlagen vergleichen. Beispiel: Sie müssen eventuell Einstellungswerte prüfen, um sicherzustellen, dass optimale Verfahren für die Umgebung eingehalten werden.

Außerdem ist ggf. ein Vergleich von Einstellungen in einer Richtlinie oder Vorlage mit den Standardeinstellungen von Citrix erforderlich.

1. Wählen Sie in Studio im linken Bereich den Knoten Richtlinie aus.
2. Klicken Sie auf die Registerkarte Vergleich und klicken Sie dann auf Auswählen. Das Dialogfeld Vorlagen und Richtlinien zum Vergleichen auswählen wird angezeigt.
3. Wählen Sie die Richtlinien und Vorlagen aus, die Sie vergleichen möchten. Aktivieren Sie das Kontrollkästchen Mit Standardeinstellungen vergleichen, um Standardwerte im Vergleich einzuschließen.
4. Klicken Sie auf Vergleichen. Die konfigurierten Einstellungen für die ausgewählten Elemente werden in Spalten angezeigt. Standardwerte (falls ausgewählt) werden standardmäßig in der zweiten Spalte angezeigt.
5. Zum Anzeigen aller Einstellungen wählen Sie Alle Einstellungen anzeigen. Wenn Sie zur Standardansicht zurückkehren möchten, in der die allen Vorlagen und Richtlinien gemeinsamen Einstellungen aufgeführt sind, wählen Sie Gemeinsame Einstellungen anzeigen.

Anwenden von Richtlinien

Sep 29, 2015

Wenn Sie eine Richtlinie bestimmten Benutzer- und Maschinenobjekten zuweisen, wird sie gemäß bestimmter Kriterien oder Regeln auf Verbindungen angewendet. Wenn keine Zuweisung hinzugefügt wurde, gilt die Richtlinie für alle Verbindungen.

Basierend auf einer Kombination von Kriterien können Sie in der Regel beliebig viele Zuweisungen für eine Richtlinie hinzufügen.

Hinweis: Sie können einer Richtlinie nur eine Citrix CloudBridge-Zuweisung hinzufügen.

In der folgenden Tabelle werden verfügbare Zuweisungen aufgelistet:

Name	Beschreibung
Zugriffssteuerung	Wendet die Richtlinie basierend auf den Zugriffssteuerungsbedingungen an, die für die Clientverbindung gelten.
Citrix CloudBridge	Wendet die Richtlinie abhängig davon an, ob der Sitzungsstart über Citrix CloudBridge erfolgte.
Client-IP-Adresse	Wendet die Richtlinie basierend auf der IP-Adresse (IPv4 oder IPv6) des Benutzergeräts an, über das die Verbindung mit der Sitzung hergestellt wird. IPv4-Beispiele: • 12.0.0.0 • 12.0.0.* • 12.0.0.1-12.0.0.70 • 12.0.0.1/24 IPv6-Beispiele: • 2001:0db8:3c4d:0015:0:0:abcd:ef12 • 2001:0db8:3c4d:0015::/54
Clientname	Wendet die Richtlinie basierend auf dem Namen des Benutzergeräts an, über das die Verbindung mit der Sitzung hergestellt wird.
Bereitstellungsgruppe	Wendet die Richtlinie basierend auf der Bereitstellungsgruppen-Mitgliedschaft des Desktops an, auf dem die Sitzung ausgeführt wird.
Desktoptyp	Wendet eine Richtlinie basierend auf dem Typ des Desktops an, auf dem die Sitzung ausgeführt wird.
Organisationseinheit	Wendet eine Richtlinie basierend auf der Organisationseinheit des Desktops an, auf dem die Sitzung ausgeführt wird.
Tag	Wendet eine Richtlinie basierend auf beliebigen Tags an, die auf dem Desktop angewendet

Name	Beschreibung
Benutzer oder Gruppe	Werden auf dem die Sitzung ausgeführt wird. Wendet die Richtlinie basierend auf dem Benutzernamen oder der Gruppenmitgliedschaft des Benutzers an, der die Verbindung mit der Sitzung herstellt.

Alle Richtlinien, die mit den Zuweisungen für die Verbindung übereinstimmen, werden bei der Anmeldung eines Benutzers identifiziert. Die identifizierten Richtlinien werden nach Priorität sortiert und mehrere Instanzen jeder Einstellung werden verglichen. Die einzelnen Einstellungen werden gemäß der Richtlinien-Prioritätsreihenfolge angewendet. Wenn Sie Active Directory verwenden, werden Richtlinieneinstellungen aktiviert, wenn Active Directory Richtlinien in 90-Minuten-Intervallen neu evaluiert, und bei der Anmeldung eines Benutzers angewendet.

Jede deaktivierte Richtlinieneinstellung hat Vorrang vor einer aktivierten Richtlinieneinstellung, deren Priorität niedriger ist. Richtlinieneinstellungen, die nicht konfiguriert sind, werden ignoriert.

Wichtig: Bei der Konfiguration von Active Directory- und Citrix Richtlinien mit der Gruppenrichtlinien-Verwaltungskonsolle werden Zuweisungen und Einstellungen möglicherweise nicht wie erwartet angewendet. Weitere Informationen finden Sie unter <http://support.citrix.com/article/CTX127461>.

Ungefilterte Richtlinien

Standardmäßig gibt es eine Richtlinie "Ungefiltert". Die Einstellungen, die der Richtlinie hinzugefügt werden, gelten für alle Verbindungen.

Wenn Sie Studio zur Verwaltung von Citrix Richtlinien verwenden, werden die Einstellungen, die Sie der Richtlinie "Ungefiltert" hinzufügen, auf alle Server, Desktops und Verbindungen einer Site angewendet.

Wenn Sie in Ihrer Umgebung Active Directory verwenden und mit dem Gruppenrichtlinien-Editor Citrix Richtlinien verwalten, gelten Einstellungen, die Sie der Richtlinie "Ungefiltert" hinzufügen, für alle Sites und Verbindungen, die zu dem Geltungsbereich des Gruppenrichtlinienobjekts gehören, das die Richtlinie enthält. Beispiel: Die Organisationseinheit (OU) "Verkauf" enthält ein Gruppenrichtlinienobjekt "Verkauf-USA", das alle Mitarbeiter des US-Verkaufsteams einschließt. Das Gruppenrichtlinienobjekt "Verkauf-USA" ist mit einer Richtlinie "Ungefiltert" konfiguriert, die mehrere Benutzerrichtlinieneinstellungen enthält. Wenn der US-Verkaufsleiter sich an der Site anmeldet, werden die Einstellungen der Richtlinie "Ungefiltert" automatisch auf die Sitzung angewendet, weil der Benutzer Mitglied des Gruppenrichtlinienobjekts "Verkauf-USA" ist.

Zuweisungsmodi

Der Modus einer Zuweisung entscheidet, ob die Richtlinie nur auf Verbindungen angewendet wird, die alle Zuweisungskriterien erfüllen. Wenn der Modus Zulassen (Standardwert) ist, wird die Richtlinie nur auf Verbindungen angewendet, die die Zuweisungskriterien erfüllen. Wenn der Modus Verweigern ist, wird die Richtlinie angewendet, wenn eine Verbindung die Zuweisungskriterien nicht erfüllt. Das folgende Beispiel zeigt, wie Zuweisungsmodi sich auf Citrix Richtlinien auswirken, wenn mehrere Zuweisungen vorhanden sind.

Beispiel: Zuweisungen des gleichen Typs mit unterschiedlichen Modi

In Richtlinien mit zwei Zuweisungen des gleichen Typs, eine mit der Einstellung Zulassen und die andere mit der Einstellung Verweigern, hat die Zuweisung mit der Einstellung Verweigern Vorrang, wenn die Verbindung die Kriterien beider Zuweisungen erfüllt. Beispiel:

Richtlinie 1 enthält die folgenden Zuweisungen:

- Zuweisung A ist eine Benutzerzuweisung, die die Verkaufsgruppe angibt, und der Modus ist Zulassen.
- Zuweisung B ist eine Benutzerzuweisung, die das Konto des Verkaufsleiters angibt, und der Modus ist Verweigern.

Da der Modus für Zuweisung B Verweigern ist, wird die Richtlinie nicht angewendet, wenn der Verkaufsleiter sich bei der Site anmeldet, obwohl er Mitglied der Verkaufsgruppe ist.

Beispiel: Zuweisungen unterschiedlichen Typs mit gleichen Modi

In Richtlinien mit zwei oder mehr Zuweisungen unterschiedlichen Typs, für die Zulassen eingestellt ist, muss die Verbindung die Kriterien von mindestens einer Zuweisung jedes Typs erfüllen, damit die Richtlinie angewendet wird. Beispiel:

Richtlinie 2 enthält die folgenden Zuweisungen:

- Zuweisung C ist eine Benutzerzuweisung, die die Verkaufsgruppe angibt, und der Modus ist Zulassen.
- Zuweisung D ist eine Client-IP-Adressenzuweisung, die 10.8.169.* festlegt (das Unternehmensnetzwerk), und der Modus ist Zulassen.

Wenn der Verkaufsleiter sich im Büro bei der Site anmeldet, wird die Richtlinie angewendet, weil die Verbindung die Kriterien beider Zuweisungen erfüllt.

Richtlinie 3 enthält die folgenden Zuweisungen:

- Zuweisung E ist eine Benutzerzuweisung, die die Verkaufsgruppe angibt, und der Modus ist Zulassen.
- Zuweisung F ist eine Zugriffssteuerungszuweisung, die NetScaler Gateway-Verbindungsbedingungen angibt, und der Modus ist Zulassen.

Wenn der Verkaufsleiter sich im Büro bei der Site anmeldet, wird die Richtlinie nicht angewendet, weil die Verbindung nicht die Kriterien von Zuweisung F erfüllt.

Anwenden einer Richtlinie

Soll eine Richtlinie nur auf bestimmte Benutzer- und Maschinenobjekte angewendet werden, müssen Sie ihr mindestens eine Zuweisung hinzufügen. Wenn Sie keine Zuweisung hinzufügen, werden die Richtlinieneinstellungen auf alle Verbindungen angewendet, es sei denn, die Richtlinieneinstellungen werden von Einstellungen einer Richtlinie mit einer höheren Priorität außer Kraft gesetzt.

1. Wählen Sie in Studio im linken Bereich den Knoten Richtlinie aus.
2. Klicken Sie auf die Registerkarte Richtlinien und wählen Sie eine bestehende Richtlinie aus oder erstellen Sie eine neue Richtlinie.
3. Befolgen Sie die Anweisungen des Richtlinien-Assistenten bis Sie zu der Seite Benutzer und Maschinen gelangen.
4. Wählen Sie die Zuweisung aus, die Sie anwenden möchten, und klicken Sie dann auf Zuweisen.
5. Wählen Sie im Dialogfeld Richtlinie zuweisen den Modus für die Zuweisung aus und konfigurieren Sie die Zuweisungselemente:

Assignment Type	Parameter
Zugriffssteuerung	• Verbindungstyp: Wählen Sie aus, ob die Richtlinie auf Verbindungen anzuwenden ist, die mit NetScaler Gateway hergestellt wurden, oder auf solche, die ohne NetScaler Gateway hergestellt wurden. • NetScaler Gateway-Farmname: Geben Sie den Namen des virtuellen Servers für NetScaler Gateway (früher "Access Gateway") ein. • Zugriffsbedingung: Geben Sie den Namen der Endpunktanalyserichtlinie oder der Sitzungsrichtlinie ein, die verwendet werden soll.

Citrix CloudBridge Assignment Type	Parameter
	Verbindungen: Wählen Sie aus, ob die Richtlinie auf Verbindungen anzuwenden ist, die mit Citrix CloudBridge gestartet wurden, oder auf solche, die ohne Citrix CloudBridge gestartet wurden.
Client-IP-Adresse	IP-Adresse: IP-Adresse des Benutzergeräts, auf das die Richtlinie angewendet werden soll.
Clientname	Clientname: Name des Benutzergeräts, auf das die Richtlinie angewendet werden soll.
Bereitstellungsgruppe	• Controller: Name des Controllers zur Verwaltung der Desktops, auf den die Richtlinie angewendet wird. • Bereitstellungsgruppe: Name der Bereitstellungsgruppe, auf die die Richtlinie angewendet wird.
Desktoptyp	Desktoptyp: Typ des Desktops, auf den Sie die Richtlinie anwenden. Wählen Sie eine der folgenden Optionen: • Privater Desktop • Freigegebener Desktop • Private Anwendung • Freigegebene Anwendung
Organisationseinheit	Organisationseinheit: Name der Organisationseinheit, auf die die Richtlinie angewendet werden soll.
Tag	• Controller: Name des Controllers zur Verwaltung der Desktops, auf den die Richtlinie angewendet wird. • Tag: Desktop-Tag, das beim Anwenden der Richtlinie gesucht werden soll.
Benutzer oder Gruppe	Benutzer- oder Gruppenname: Name des Benutzers bzw. der Gruppe, auf den bzw. die die Richtlinie angewendet werden soll.

Die Richtlinie wird angewendet, wenn die relevanten Benutzer das nächste Mal eine Verbindung herstellen.

Verwenden mehrerer Richtlinien für die Anpassung von Benutzerzugriff

Aktualisiert: 2015-04-26

Sie können mehrere Richtlinien verwenden, um Ihre Umgebung an die Anforderungen der Benutzer, basierend auf deren Aufgabengebiet, geografischem Standort oder Verbindungstyp anzupassen. Beispielsweise sind Sie vielleicht aus Sicherheitsgründen gezwungen, Benutzergruppen, die regelmäßig mit äußerst wichtigen Daten Umgang haben, Beschränkungen aufzuerlegen. Sie können eine Richtlinie erstellen, die Benutzer daran hindert, vertrauliche Daten auf ihren lokalen Clientlaufwerken zu speichern. Wenn jedoch manche Mitglieder dieser Benutzergruppe Zugang zu ihren lokalen Laufwerken benötigen, können Sie eine andere Richtlinie für diese Benutzer erstellen. Anschließend können Sie den beiden Richtlinien jeweils eine Priorität zuweisen und damit festlegen, welche Richtlinie Vorrang haben soll.

Wenn Sie mehrere Richtlinien verwenden, müssen Sie festlegen, wie Prioritäten zugewiesen und Ausnahmen erstellt werden und wie die wirksame Richtlinie bei Richtlinienkonflikten angezeigt wird.

In der Regel setzen Richtlinien ähnliche Einstellungen, die für die gesamte Site, für bestimmte Controller oder auf dem

Benutzergerät konfiguriert wurden, außer Kraft. Die Ausnahme von diesem Prinzip sind Sicherheitseinstellungen. Die höchste Verschlüsselungseinstellung in der Umgebung, einschließlich Betriebssystem, und die Spiegelungseinstellung mit der größten Einschränkung haben immer Vorrang vor allen anderen Einstellungen und Richtlinien.

Citrix Richtlinien interagieren mit den Richtlinien, die Sie im Betriebssystem eingestellt haben. In einer Citrix Umgebung überschreiben Citrix Einstellungen die gleichen Einstellungen in einer Active Directory-Richtlinie oder in der Konfiguration des Remotedesktop-Sitzungshosts. Dazu gehören auch Einstellungen, die mit typischen Remotedesktopprotokoll-Clientverbindungseinstellungen zusammenhängen, wie Desktophintergrund, Menüanimation und das Anzeigeverhalten beim Drag & Drop. Manche Richtlinieneinstellungen, wie SecureICA, müssen mit den Richtlinien und Einstellungen im Betriebssystem übereinstimmen. Wenn anderswo ein höherer Verschlüsselungsgrad festgelegt wurde, kann die SecureICA-Richtlinieneinstellung oder die Einstellung beim Veröffentlichen einer Anwendung außer Kraft gesetzt werden.

Beispiel: Verschlüsselungseinstellungen, die Sie beim Erstellen von Bereitstellungsgruppen zur Bereitstellung von Anwendungen und Desktops für Benutzer angeben, müssen mindestens den gleichen Verschlüsselungsgrad aufweisen wie die umgebungsweit geltenden Verschlüsselungseinstellungen.

Festlegen der Richtlinienpriorität und Erstellen von Ausnahmen

Durch Festlegen der Richtlinienpriorität definieren Sie, welche Richtlinie Vorrang hat, wenn es Konflikte gibt. Der Prozess für die Auswertung der Richtlinien ist wie folgt:

1. Alle Richtlinien, die mit den Zuweisungen für die Verbindung übereinstimmen, werden bei der Anmeldung eines Benutzers identifiziert.
2. Die identifizierten Richtlinien werden nach Priorität sortiert und mehrere Instanzen jeder Einstellung werden verglichen. Die einzelnen Einstellungen werden gemäß der Richtlinien-Prioritätsreihenfolge angewendet.

Sie weisen Richtlinien eine Priorität zu, indem Sie ihnen eine andere Prioritätszahl geben. Neue Richtlinien erhalten standardmäßig die niedrigste Priorität. Falls widersprüchliche Richtlinieneinstellungen auftreten, setzt eine Richtlinie mit einem höheren Prioritätswert (eine Priorität von "1" hat die höchste Priorität) eine Richtlinie mit einem niedrigeren Prioritätswert außer Kraft. Einstellungen werden nach der Priorität und dem Zustand der Einstellung, z. B. ob die Einstellung deaktiviert oder aktiviert ist, zusammengeführt. Jede deaktivierte Einstellung hat Vorrang vor einer aktivierten Einstellung, deren Priorität niedriger ist. Richtlinieneinstellungen, die nicht konfiguriert sind, werden ignoriert und setzen keine Einstellungen mit niedrigerer Priorität außer Kraft.

Wenn Sie Richtlinien für Benutzergruppen, Benutzergeräte oder Maschinen erstellen, werden Sie möglicherweise feststellen, dass für einige Mitglieder einer Gruppe Ausnahmen zu einigen Einstellungen erstellt werden müssen. Sie können Ausnahmen wie folgt erstellen:

- Erstellen Sie eine Richtlinie für die Gruppenmitglieder, für die Ausnahmen erforderlich sind, und stufen Sie die Richtlinie mit höherer Priorität ein als die Richtlinie für die gesamte Gruppe.
- Verwenden Sie den Modus Verweigern in einer Zuweisung, die Sie der Richtlinie hinzufügen.

Die Zuweisung im Modus Verweigern wendet eine Richtlinie nur auf Verbindungen an, die nicht den Zuweisungskriterien entsprechen. Beispielsweise könnte eine Richtlinie folgende Zuweisungen enthalten:

- Zuweisung A ist eine Client-IP-Adressenzuweisung, die den Bereich 208.77.88.* festlegt, und der Modus ist Zulassen.
- Zuweisung B ist eine Benutzerzuweisung, die ein spezifisches Benutzerkonto angibt, und der Modus ist Verweigern.

Die Richtlinie wird auf alle Benutzer angewendet, die sich mit IP-Adressen aus dem in Zuweisung A festgelegten Bereich bei der Site anmelden. Die Richtlinie wird aber nicht auf den Benutzer angewendet, der sich mit dem in Zuweisung B festgelegten Konto anmeldet, obwohl dem Computer dieses Benutzers eine IP-Adresse aus dem in Zuweisung A festgelegten Bereich zugewiesen wurde.

Ermitteln der auf eine Verbindung angewendeten Richtlinien

Sep 29, 2015

Manchmal reagiert eine Verbindung nicht wie erwartet, weil mehrere Richtlinien gelten. Wenn eine Richtlinie mit einer höheren Priorität auch auf eine Verbindung angewendet wird, kann sie Einstellungen, die Sie in der ursprünglichen Richtlinie konfigurieren, außer Kraft setzen. Sie können ermitteln, wie die Richtlinieneinstellungen am Ende für eine Verbindung zusammengeführt werden, indem sie den Richtlinienergebnissatz berechnen.

Sie berechnen den Richtlinienergebnissatz mit folgenden Methoden:

- Verwenden Sie den Assistenten für die Citrix Gruppenrichtlinienmodellierung, um ein Verbindungsszenario zu simulieren und festzustellen, wie Citrix Richtlinien angewendet werden können.
- Verwenden Sie das Tool "Gruppenrichtlinienergebnisse", um einen Bericht zu erstellen, der beschreibt, welche Citrix Richtlinien für einen bestimmten Benutzer oder einen bestimmten Controller angewendet werden.

Sie können den Assistenten für die Citrix Gruppenrichtlinienmodellierung im Bereich "Aktionen" in Studio starten. Wenn Ihre Umgebung Active Directory enthält, können Sie beide Tools über die Gruppenrichtlinien-Verwaltungskonsolle in Windows starten.

Modellieren der Citrix Gruppenrichtlinie

Mit dem Assistenten für die Citrix Gruppenrichtlinienmodellierung können Sie Bedingungen für ein Verbindungsszenario angeben, z. B. Domänencontroller, Benutzer, Citrix Richtlinienzuweisungsbewertungen und simulierte Umgebungseinstellungen, wie etwa langsame Netzwerkverbindungen. Der von dem Assistenten erstellte Bericht listet die Citrix Richtlinien auf, die in dem Szenario wahrscheinlich wirksam werden.

Wenn Sie beim Controller als Domänenbenutzer angemeldet sind und die Umgebung Active Directory enthält, berechnet der Assistent den Richtlinienergebnissatz anhand von Richtlinieneinstellungen für die Site und Active Directory-Gruppenrichtlinienobjekte.

Überprüfen der Gruppenrichtlinienergebnisse

Das Tool "Gruppenrichtlinienergebnisse" unterstützt Sie dabei, den aktuellen Zustand von Gruppenrichtlinienobjekten in der Umgebung zu evaluieren, und generiert einen Bericht, in dem beschrieben wird, wie diese Objekte, einschließlich Citrix Richtlinien, derzeit auf einen bestimmten Benutzer und Controller angewendet werden.

Festlegen des zu verwendenden Richtlinienmodellierungstools

Wenn Sie den Assistenten für die Citrix Gruppenrichtlinienmodellierung oder das Tool "Gruppenrichtlinienergebnisse" über die Gruppenrichtlinien-Verwaltungskonsolle ausführen, werden die mit Studio erstellten Site-Richtlinieneinstellungen nicht in den Richtlinienergebnissatz einbezogen.

Um sicherzustellen, dass Sie den umfassendsten Richtlinienergebnissatz erhalten, empfiehlt Citrix das Starten des Assistenten für die Citrix Gruppenrichtlinienmodellierung über Studio, es sei denn, Sie erstellen Richtlinien nur über die Gruppenrichtlinien-Verwaltungskonsolle.

Problembehandlung bei Richtlinien

Für Benutzer, IP-Adressen und andere zugewiesene Objekte können mehrere Richtlinien gleichzeitig gelten. Dies kann zu

Konflikten führen, wenn eine Richtlinie sich nicht wie erwartet verhält. Wenn Sie den Citrix Gruppenrichtlinienmodellierungsassistenten oder das Gruppenrichtlinienergebnisse-Tool ausführen, entdecken Sie möglicherweise, dass keine Richtlinien auf die Benutzerverbindungen angewendet werden. In diesen Fall sind Benutzer nicht von Richtlinieneinstellungen betroffen, wenn sie sich unter Bedingungen mit Anwendungen verbinden, die den Richtlinienkriterien entsprechen. Dies passiert in folgenden Fällen:

- Keine Richtlinie hat eine Zuweisung, die den Richtlinienkriterien entspricht.
- Richtlinien, die der Zuweisung entsprechen, haben keine konfigurierten Einstellungen.
- Richtlinien, die der Zuweisung entsprechen, sind deaktiviert.

Wenn Sie Richtlinieneinstellungen auf Verbindungen anwenden möchten, die bestimmten Kriterien entsprechen:

- Stellen Sie sicher, dass die Richtlinien aktiviert sind, die auf diese Verbindungen angewendet werden
- Stellen Sie sicher, dass in den Richtlinien, die auf diese Verbindungen angewendet werden, die geeigneten Einstellungen konfiguriert sind

Simulieren von Verbindungsszenarios mit Citrix Richtlinien

1. Abhängig von der Umgebung, öffnen Sie den Assistenten für die Citrix Gruppenrichtlinienmodellierung wie folgt:
 - Wählen Sie in Studio den Knoten Richtlinie im linken Bereich der Konsole und klicken Sie dann auf die Registerkarte Modellierung. Wählen Sie im Bereich "Aktionen" die Option Modellierungsassistenten starten aus.
 - Klicken Sie in der Gruppenrichtlinien-Verwaltungskontrolle mit der rechten Maustaste in der Konsolenstruktur auf den Knoten Citrix Gruppenrichtlinienmodellierung und wählen Sie Citrix Gruppenrichtlinienmodellierungsassistent aus.
2. Folgen Sie dem Assistenten, um den Domänencontroller, Benutzer, Computer, Umgebungseinstellungen und Citrix Zuweisungskriterien auszuwählen, die Sie in der Simulation verwenden möchten.

Wenn Sie auf Fertig stellen klicken, erstellt der Assistent einen Bericht mit den Modellierungsergebnissen. In Studio wird der Bericht im mittleren Bereich unter der Registerkarte Modellierung angezeigt.

Zum Anzeigen des Berichts wählen Sie Modellierungsbericht anzeigen. In dem Bericht sind effektive Citrix Richtlinieneinstellungen unter den Überschriften "Citrix Computerrichtlinien" und "Citrix Benutzerrichtlinien" zusammengefasst.

Verwalten von Richtlinien mit dem Gruppenrichtlinien-Editor

Sep 29, 2015

Wenn Sie in Ihrer Netzwerkumgebung Active Directory verwenden und Sie die notwendigen Berechtigungen zur Verwaltung von Gruppenrichtlinien haben, können Sie den Gruppenrichtlinien-Editor verwenden, um Richtlinien für Ihre Site zu erstellen. Die Einstellungen, die Sie konfigurieren, beeinträchtigen die Gruppenrichtlinienobjekte, die Sie in der Gruppenrichtlinien-Verwaltungskonsolle angeben.

Wichtig: Zum Konfigurieren bestimmter Einstellungen müssen Sie den Gruppenrichtlinien-Editor verwenden, z. B. für Einstellungen, die sich auf die Registrierung virtueller Desktops bei einem Controller beziehen.

Vollständige Anweisungen zur Steuerung von Richtlinien mit dem Gruppenrichtlinien-Editor finden Sie unter

— *Arbeiten mit Richtlinien*

in der Dokumentation für [XenDesktop 5](#) in eDocs.

Erstellen und Konfigurieren von Richtlinien und Vorlagen

1. Erweitern Sie im Gruppenrichtlinien-Editor den Knoten Computerkonfiguration oder Benutzerkonfiguration, erweitern Sie den Knoten Richtlinien und wählen Sie dann Citrix Richtlinien.
2. Führen Sie einen der folgenden Schritte aus:

Aufgabe	Anweisungen
Erstellen einer neuen Richtlinie	Klicken Sie auf der Registerkarte Richtlinien auf Neu.
Bearbeiten einer bestehenden Richtlinie	Wählen Sie auf der Registerkarte Richtlinien die Richtlinie aus, die Sie bearbeiten möchten, und klicken Sie dann auf Bearbeiten.
Ändern der Priorität einer Richtlinie	Wählen Sie auf der Registerkarte Richtlinien die Richtlinie aus, für die Sie die Priorität ändern möchten, und klicken Sie dann auf Höher oder Niedriger.
Anzeigen einer Zusammenfassung über eine Richtlinie	Wählen Sie auf der Registerkarte Richtlinien die gewünschte Richtlinie aus und klicken Sie dann auf die Registerkarte Zusammenfassung.
Anzeigen und Ergänzen der Einstellungen einer Richtlinie	Wählen Sie auf der Registerkarte Richtlinien die gewünschte Richtlinie aus und klicken Sie dann auf die Registerkarte Einstellungen.
Anzeigen und Ergänzen der Filter für eine Richtlinie	Wählen Sie auf der Registerkarte Richtlinien die gewünschte Richtlinie aus und klicken Sie dann auf die Registerkarte Filter.
Aktivieren oder Deaktivieren einer Richtlinie	Wählen Sie auf der Registerkarte Richtlinien die gewünschte Richtlinie aus und wählen Sie Aktionen > Aktivieren oder Aktionen > Deaktivieren.

Aufgabe Erstellen einer Vorlage auf der Basis einer vorhandenen Richtlinie	Anweisungen Wählen Sie auf der Registerkarte Richtlinien die Richtlinie aus, auf deren Basis Sie eine Vorlage erstellen möchten, und wählen Sie dann Aktionen > Als Vorlage speichern.
Erstellen einer Richtlinie basierend auf einer vorhandenen Vorlage	Wählen Sie auf der Registerkarte Richtlinien die Vorlage aus, auf deren Basis Sie eine Richtlinie erstellen möchten, und wählen Sie dann Neue Richtlinie.
Erstellen einer Vorlage basierend auf einer vorhandenen Vorlage	Wählen Sie auf der Registerkarte Richtlinien die Vorlage aus, auf deren Basis Sie eine Vorlage erstellen möchten, und wählen Sie dann Neue Vorlage.
Importieren einer Vorlage	Wählen Sie auf der Registerkarte Richtlinien Aktionen > Importieren.
Exportieren einer Vorlage	Wählen Sie auf der Registerkarte Richtlinien Aktionen > Exportieren.
Anzeigen der Vorlageneinstellungen	Wählen Sie auf der Registerkarte Vorlage die gewünschte Vorlage aus und klicken Sie dann auf die Registerkarte Einstellungen.
Anzeigen einer Zusammenfassung von Vorlageneigenschaften	Wählen Sie auf der Registerkarte Vorlage die gewünschte Vorlage aus und klicken Sie dann auf die Registerkarte Eigenschaften.
Anzeigen von Vorlagenvoraussetzungen	Wählen Sie auf der Registerkarte Vorlage die gewünschte Vorlage aus und klicken Sie dann auf die Registerkarte Voraussetzungen.

Standardrichtlinieneinstellungen

Sep 29, 2015

ICA

Name	Standardeinstellung	Geltungsbereich
Clientzwischenablagenumleitung	Zugelassen	Alle Virtual Delivery Agent-Versionen
Desktop starten	Nicht zugelassen	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)
ICA-Listener - Verbindungszeitout	120000 Millisekunden	Virtual Delivery Agent 5, 5.5, 5.6 Feature Pack 1, 7 (Desktopbetriebssystem), 7.1 (Desktopbetriebssystem), 7.5 (Desktopbetriebssystem)
ICA-Listenerportnummer	1494	Alle Virtual Delivery Agent-Versionen
Starten nicht-veröffentlichter Programme bei Clientverbindung	Nicht zugelassen	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)
Lesezugriff auf Zwischenablage	Nicht zugelassen	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)

ICA/Audio

Name	Standardeinstellung	Geltungsbereich
Audio Plug & Play	Zugelassen	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)
Audioqualität	Hoch - High Definition Audio	Alle Virtual Delivery Agent-Versionen
Clientaudioumleitung	Zugelassen	Alle Virtual Delivery Agent-Versionen
Clientmikrofonumleitung	Zugelassen	Alle Virtual Delivery Agent-Versionen

ICA/Automatische Wiederverbindung von Clients

Name	Standardeinstellung	Geltungsbereich
Automatische Wiederverbindung von Clients	Zugelassen	Alle Virtual Delivery Agent-Versionen
Authentifizierung bei automatischer Wiederverbindung von Clients	Keine Authentifizierung erforderlich	Alle Virtual Delivery Agent-Versionen
Protokollierung der automatischen Wiederverbindung von Clients	Kein Protokollieren von Wiederverbindungsereignissen	Alle Virtual Delivery Agent-Versionen

ICA/Bandwidth

Name	Standardeinstellung	Geltungsbereich
Bandbreitenlimit für die Audioumleitung	0 KBit/s	Alle Virtual Delivery Agent-Versionen
Bandbreitenlimit für die Audioumleitung (Prozent)	0	Alle Virtual Delivery Agent-Versionen
Bandbreitenlimit für Client-USB-Geräteumleitung	0 KBit/s	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Bandbreitenlimit für Client-USB-Geräteumleitung (Prozent)	0	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Bandbreitenlimit für Zwischenablagenumleitung	0 KBit/s	Alle Virtual Delivery Agent-Versionen
Bandbreitenlimit für Zwischenablagenumleitung (Prozent)	0	Alle Virtual Delivery Agent-Versionen
Bandbreitenlimit für COM-	0 KBit/s	Alle Virtual Delivery Agent-Versionen

Portumleitung Name	Standardeinstellung	Geltungsbereich
		Hinweis: Konfigurieren Sie diese Einstellung für Virtual Delivery Agent 7.x über die Registrierung.
Bandbreitenlimit für COM-Portumleitung (Prozent)	0	Alle Virtual Delivery Agent-Versionen Hinweis: Konfigurieren Sie diese Einstellung für Virtual Delivery Agent 7.x über die Registrierung.
Bandbreitenlimit für Dateiumleitung	0 KBit/s	Alle Virtual Delivery Agent-Versionen
Bandbreitenlimit für Dateiumleitung (Prozent)	0	Alle Virtual Delivery Agent-Versionen
Bandbreitenlimit für HDX MediaStream-Multimediabeschleunigung	0 KBit/s	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Bandbreitenlimit für HDX MediaStream-Multimediabeschleunigung (Prozent)	0	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Bandbreitenlimit für LPT-Portumleitung	0 KBit/s	Alle Virtual Delivery Agent-Versionen Hinweis: Konfigurieren Sie diese Einstellung für Virtual Delivery Agent 7.x über die Registrierung.
Bandbreitenlimit für LPT-Portumleitung (Prozent)	0	Alle Virtual Delivery Agent-Versionen Hinweis: Konfigurieren Sie diese Einstellung für Virtual Delivery Agent 7.x über die Registrierung.
Bandbreitenlimit für Sitzung insgesamt	0 KBit/s	Alle Virtual Delivery Agent-Versionen
Bandbreitenlimit für Druckerumleitung	0 KBit/s	Alle Virtual Delivery Agent-Versionen
Bandbreitenlimit für Druckerumleitung (Prozent)	0	Alle Virtual Delivery Agent-Versionen
Bandbreitenlimit für TWAIN-Geräteumleitung	0 KBit/s	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1

Name	Standardeinstellung	Geltungsbereich
		(Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Bandbreitenlimit für TWAIN-Geräteumleitung (Prozent)	0	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)

ICA/Clientsensoren

Name	Standardeinstellung	Geltungsbereich
Anwendungen können den physischen Standort des Clientgeräts verwenden	Nicht zugelassen	Virtual Delivery Agent 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)

ICA/Desktopbenutzeroberfläche

Name	Standardeinstellung	Geltungsbereich
Desktopgestaltungsumleitung	Aktiviert	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Desktopbetriebssystem), 7.1 (Desktopbetriebssystem), 7.5 (Desktopbetriebssystem)
Grafikqualität Desktopgestaltung	Mittel	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Desktopbetriebssystem), 7.1 (Desktopbetriebssystem), 7.5 (Desktopbetriebssystem)
Desktophintergrund	Zugelassen	Alle Virtual Delivery Agent-Versionen
Menüanimation	Zugelassen	Alle Virtual Delivery Agent-Versionen
Fensterinhalt beim Verschieben anzeigen	Zugelassen	Alle Virtual Delivery Agent-Versionen

ICA/Endbenutzerüberwachung

Name	Standardeinstellung	Geltungsbereich
ICA-Roundtripberechnung	Aktiviert	Alle Virtual Delivery Agent-Versionen

Name	Standardeinstellung	Geltungsbereich
Intervall für ICA-Roundtripberechnung	15 Sekunden	Alle Virtual Delivery Agent-Versionen
ICA-Roundtrip für Verbindungen im Leerlauf berechnen	Deaktiviert	Alle Virtual Delivery Agent-Versionen

ICA/Enhanced Desktop Experience

Name	Standardeinstellung	Geltungsbereich
Enhanced Desktop Experience	Zugelassen	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)

ICA/Dateiumleitung

Name	Standardeinstellung	Geltungsbereich
Clientlaufwerke automatisch verbinden	Zugelassen	Alle Virtual Delivery Agent-Versionen
Clientlaufwerkumleitung	Zugelassen	Alle Virtual Delivery Agent-Versionen
Lokale Clientfestplattenlaufwerke	Zugelassen	Alle Virtual Delivery Agent-Versionen
Clientdiskettenlaufwerke	Zugelassen	Alle Virtual Delivery Agent-Versionen
Clientnetzlaufwerke	Zugelassen	Alle Virtual Delivery Agent-Versionen
Optische Clientlaufwerke	Zugelassen	Alle Virtual Delivery Agent-Versionen
Clientwechsellaufwerke	Zugelassen	Alle Virtual Delivery Agent-Versionen
Host-zu-Client-Umleitung	Deaktiviert	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)
Clientlaufwerksbuchstaben erhalten	Deaktiviert	Virtual Delivery Agent 5, 5.5, 5.6 Feature Pack 1, 7 (Desktopbetriebssystem), 7.1 (Desktopbetriebssystem), 7.5 (Desktopbetriebssystem)

Schreibgeschützter Zugriff auf Clientlaufwerke	Deaktiviert	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Umleitung spezieller Ordner	Zugelassen	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem) Nur Webinterface-Bereitstellungen
Asynchrones Schreiben verwenden	Deaktiviert	Alle Virtual Delivery Agent-Versionen

ICA/Graphics

Name	Standardeinstellung	Geltungsbereich
Anzeigespeicherlimit	65536 KBit	Virtual Delivery Agent 5, 5.5, 5.6 Feature Pack 1, 7 (Desktopbetriebssystem), 7.1 (Desktopbetriebssystem), 7.5 (Desktopbetriebssystem)
Herabsetzungspräferenz für Anzeigemodus	Zuerst Farbtiefe herabsetzen	Alle Virtual Delivery Agent-Versionen
Dynamische Fenstervorschau	Aktiviert	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Bildzwischenspeicherung	Aktiviert	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Legacygrafikmodus	Deaktiviert	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Maximal zugelassene Farbtiefe	32 Bit pro Pixel	Alle Virtual Delivery Agent-Versionen

Benutzer beim Herabsetzen des Anzeigemodus benachrichtigen	Deaktiviert	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)
Warteschlange und Verwerfen	Aktiviert	Alle Virtual Delivery Agent-Versionen

ICA/Graphics/Caching

Name	Standardeinstellung	Geltungsbereich
Schwellenwert für permanenten Cache	3000000 Bit/s	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)

ICA/Keep-Alive

Name	Standardeinstellung	Geltungsbereich
ICA-Keep-Alive - Timeout	60 Sekunden	Alle Virtual Delivery Agent-Versionen
ICA-Keep-Alives	Keine ICA-Keep-Alive-Meldungen senden	Alle Virtual Delivery Agent-Versionen

ICA/Zugriff auf lokale Anwendungen

Name	Standardeinstellung	Geltungsbereich
Lokalen App-Zugriff zulassen	Nicht zugelassen	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
URL-Umleitungssperrliste	Keine Sites angegeben	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
URL-Umleitungspositivliste	Keine Sites angegeben	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)

Name	Standardeinstellung	Geltungsbereich
ICA/Mobilerfahrung		

Name	Standardeinstellung	Geltungsbereich
Automatische Anzeige der Tastatur	Nicht zugelassen	Virtual Delivery Agent 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Für Fingereingabe optimierten Desktop starten	Zugelassen	Virtual Delivery Agent 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Kombinationsfelder remoten	Nicht zugelassen	Virtual Delivery Agent 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)

ICA/Multimedia

Name	Standardeinstellung	Geltungsbereich
Videoqualität beschränken	Nicht konfiguriert	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Multimediakonferenzen	Zugelassen	Alle Virtual Delivery Agent-Versionen
Optimierung von Windows Media-Multimediaumleitung über WAN	Zugelassen	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
GPU für die Optimierung von Windows Media-Multimediaumleitung über WAN verwenden	Nicht zugelassen	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Clientseitiger Abruf von Windows	Zugelassen	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem)

Media-Inhalten Name	Standardeinstellung	Geltungsbereich
		Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Windows Media-Umleitung	Zugelassen	Alle Virtual Delivery Agent-Versionen
Windows Media-Umleitungspuffergröße	5 Sekunden	Virtual Delivery Agent 5, 5.5 und 5.6, Feature Pack 1
Verwendung von Windows Media-Umleitungspuffergröße	Deaktiviert	Virtual Delivery Agent 5, 5.5 und 5.6, Feature Pack 1

ICA/Multistreamverbindungen

Name	Standardeinstellung	Geltungsbereich
Audio über UDP	Zugelassen	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)
Audio-UDP-Portbereich	16500, 16509	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Multiportrichtlinie	Primärer Port (2598) hat hohe Priorität	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Multistreamcomputereinstellung	Deaktiviert	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Multistreambenutzereinstellung	Deaktiviert	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)

ICA\Portumleitung

Name	Standardeinstellung	Geltungsbereich
Client-COM-Ports automatisch verbinden	Deaktiviert	Alle Virtual Delivery Agent-Versionen Hinweis: Konfigurieren Sie diese Einstellung für Virtual Delivery Agent 7.x über die Registrierung.
Client-LPT-Ports automatisch verbinden	Deaktiviert	Alle Virtual Delivery Agent-Versionen Hinweis: Konfigurieren Sie diese Einstellung für Virtual Delivery Agent 7.x über die Registrierung.
Client-COM-Portumleitung	Nicht zugelassen	Alle Virtual Delivery Agent-Versionen Hinweis: Konfigurieren Sie diese Einstellung für Virtual Delivery Agent 7.x über die Registrierung.
Client-LPT-Portumleitung	Nicht zugelassen	Alle Virtual Delivery Agent-Versionen Hinweis: Konfigurieren Sie diese Einstellung für Virtual Delivery Agent 7.x über die Registrierung.

ICA/Printing

Name	Standardeinstellung	Geltungsbereich
Clientdruckerumleitung	Zugelassen	Alle Virtual Delivery Agent-Versionen
Standarddrucker	Hauptdrucker des Clients als Standarddrucker verwenden	Alle Virtual Delivery Agent-Versionen
Druckerzuweisungen	Der aktuelle Drucker des Benutzers wird als Standarddrucker in der Sitzung verwendet.	Alle Virtual Delivery Agent-Versionen
Ereignisprotokollpräferenz für die automatische Druckererstellung	Fehler und Warnungen protokollieren	Alle Virtual Delivery Agent-Versionen
Sitzungsdrucker	Keine Drucker angegeben	Alle Virtual Delivery Agent-Versionen
Warten bis Drucker erstellt sind (Desktop)	Deaktiviert	Alle Virtual Delivery Agent-Versionen

ICA/Drucken/Clientdrucker

Name	Standardeinstellung	Geltungsbereich
Clientdrucker automatisch erstellen	Alle Clientdrucker automatisch erstellen	Alle Virtual Delivery Agent - Versionen
Generischen universellen Drucker automatisch erstellen	Deaktiviert	Alle Virtual Delivery Agent - Versionen
Clientdruckernamen	Standarddruckernamen	Alle Virtual Delivery Agent - Versionen
Direkte Verbindungen zu Druckservern	Aktiviert	Alle Virtual Delivery Agent - Versionen
Druckertreiberzuordnung und -kompatibilität	Keine Regeln angegeben	Alle Virtual Delivery Agent - Versionen
Speicherung von Druckereigenschaften	Im Profil speichern, wenn sie nicht auf dem Client gespeichert sind	Alle Virtual Delivery Agent - Versionen
Gespeicherte und wiederhergestellte Clientdrucker	Zugelassen	Virtual Delivery Agent 5, 5.5 und 5.6, Feature Pack 1

ICA/Printing/Drivers

Name	Standardeinstellung	Geltungsbereich
Automatische Installation von mitgelieferten Druckertreibern	Aktiviert	Alle Virtual Delivery Agent-Versionen
Priorität universeller Treiber	EMF, XPS, PCL5c, PCL4, PS	Alle Virtual Delivery Agent-Versionen
Verwendung universeller Druckertreiber	Universelles Drucken nur verwenden, wenn angeforderter Treiber nicht verfügbar ist	Alle Virtual Delivery Agent-Versionen

ICA/Drucken/Universeller Druckserver

Name	Standardeinstellung	Geltungsbereich
Universellen Druckserver aktivieren	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Port für Druckdatenstrom des universellen Druckservers (CGP)	7229	Alle Virtual Delivery Agent-Versionen
Universeller Druckserver - Eingabebandbreitenlimit für Druckdatenstrom (KBit/s)	0	Alle Virtual Delivery Agent-Versionen
Port für universellen Druckserverwebdienst (HTTP/SOAP)	8080	Alle Virtual Delivery Agent-Versionen

ICA/Drucken/Universelles Drucken

Name	Standardeinstellung	Geltungsbereich
Universelles Drucken - EMF-Verarbeitungsmodus	Direkt zum Drucker spoolen	Alle Virtual Delivery Agent-Versionen
Universelles Drucken - Bildkomprimierungslimit	Beste Qualität (verlustfreie Komprimierung)	Alle Virtual Delivery Agent-Versionen
Universelles Drucken - Optimierungsstandards	Bildkomprimierung • Gewünschte Bildqualität=Standardqualität • Heavyweight-Komprimierung aktivieren=False Zwischenspeichern von Bildern und Schriftarten • Zwischenspeichern eingebetteter Bilder zulassen=True • Zwischenspeichern eingebetteter Schriftarten zulassen=True Nicht-Administratoren können diese Einstellungen anpassen=False	Alle Virtual Delivery Agent-Versionen
Universelles Drucken - VorschauEinstellung	Druckvorschau für automatisch erstellte oder generische universelle Drucker nicht verwenden	Alle Virtual Delivery Agent-Versionen
Universelles Drucken - Druckqualitätslimit	Kein Limit	Alle Virtual Delivery Agent-Versionen

Name	Standardeinstellung	Geltungsbereich
ICA/Security		
Name	Standardeinstellung	Geltungsbereich
SecureICA-Mindestverschlüsselungsgrad	Standard	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)

ICA/Serverlimits

Name	Standardeinstellung	Geltungsbereich
Serverleerlauf-Zeitintervall	0 Millisekunden	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)

ICA/Sitzungslimits

Name	Standardeinstellung	Geltungsbereich
Timer für getrennte Sitzung	Deaktiviert	Virtual Delivery Agent 5, 5.5, 5.6 Feature Pack 1, 7 (Desktopbetriebssystem), 7.1 (Desktopbetriebssystem), 7.5 (Desktopbetriebssystem)
Getrennte Sitzungen - Timerintervall	1440 Minuten	Virtual Delivery Agent 5, 5.5, 5.6 Feature Pack 1, 7 (Desktopbetriebssystem), 7.1 (Desktopbetriebssystem), 7.5 (Desktopbetriebssystem)
Sitzungsverbindungstimer	Deaktiviert	Virtual Delivery Agent 5, 5.5, 5.6 Feature Pack 1, 7 (Desktopbetriebssystem), 7.1 (Desktopbetriebssystem), 7.5 (Desktopbetriebssystem)
Sitzungsverbindung - Timerintervall	1440 Minuten	Virtual Delivery Agent 5, 5.5, 5.6 Feature Pack 1, 7 (Desktopbetriebssystem), 7.1 (Desktopbetriebssystem), 7.5 (Desktopbetriebssystem)
Sitzungsleerlauf-timer	Enabledf	Virtual Delivery Agent 5, 5.5, 5.6 Feature Pack 1, 7 (Desktopbetriebssystem), 7.1 (Desktopbetriebssystem), 7.5 (Desktopbetriebssystem)
Sitzungsleerlauf - Timerintervall	1440 Minuten	Virtual Delivery Agent 5, 5.5, 5.6 Feature Pack 1, 7 (Desktopbetriebssystem), 7.1 (Desktopbetriebssystem), 7.5 (Desktopbetriebssystem)

Name	Standardeinstellung	(Desktopbetriebssystem) Geltungsbereich
------	---------------------	--

ICA/Sitzungszuverlässigkeit

Name	Standardeinstellung	Geltungsbereich
Sitzungszuverlässigkeit - Verbindungen	Zugelassen	Alle Virtual Delivery Agent-Versionen
Sitzungszuverlässigkeit - Portnummer	2598	Alle Virtual Delivery Agent-Versionen
Sitzungszuverlässigkeit - Timeout	180 Sekunden	Alle Virtual Delivery Agent-Versionen

ICA/Zeitzonesteuerung

Name	Standardeinstellung	Geltungsbereich
Lokale Zeitzone für Legacyclients schätzen	Aktiviert	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)
Lokale Zeit des Clients verwenden	Serverzeitzone verwenden	Alle Virtual Delivery Agent-Versionen

ICA/TWAIN-Geräte

Name	Standardeinstellung	Geltungsbereich
TWAIN-Geräteumleitung für Client	Zugelassen	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
TWAIN-Komprimierungsgrad	Mittel	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)

ICA/USB-Geräte

Name	Standardeinstellung	Geltungsbereich

Name	Standardeinstellung	Geltungsbereich
Client-USB-Geräteumleitung	Nicht zugelassen	Alle Virtual Delivery Agent-Versionen
Regeln für die Client-USB-Geräteumleitung	Keine Regeln angegeben	Alle Virtual Delivery Agent-Versionen
Client-USB-Geräteumleitung für Plug & Play-Geräte	Zugelassen	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)

ICA/Visuelle Anzeige

Name	Standardeinstellung	Geltungsbereich
Frameratesollwert	30 F/s	Alle Virtual Delivery Agent-Versionen
Bildqualität	Mittel	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)

ICA/Visuelle Anzeige/Bewegtbilder

Name	Standardeinstellung	Geltungsbereich
Mindestbildqualität	Normal	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Bewegtbildkomprimierung	Aktiviert	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Grad der progressiven Komprimierung	Keine	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)

Name	Standard-einstellung	Geltungsbereich
Schwellenwert für progressive Komprimierung	8192 KBit/s	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Mindestframeratesollwert	10 F/s	Virtual Delivery Agent 5.5, 5.6 Feature Pack 1, 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)

ICA/Visuelle Anzeige/Festbilder

Name	Standard-einstellung	Geltungsbereich
Zusätzliche Farbkomprimierung	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Schwellenwert für zusätzliche Farbkomprimierung	8192 KBit/s	Alle Virtual Delivery Agent-Versionen
Heavyweight-Komprimierung	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Grad der verlustreichen Komprimierung	Mittel	Alle Virtual Delivery Agent-Versionen
Schwellenwert für verlustreiche Komprimierung	2147483647 KBit/s	Alle Virtual Delivery Agent-Versionen

ICA/WebSockets

Name	Standard-einstellung	Geltungsbereich
WebSockets-Verbindungen	Nicht zugelassen	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
WebSockets-Portnummer	8008	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Vertrauenswürdige WebSockets-Ursprungsserverliste	Bei Verwendung des Platzhalters * wird allen Receiver für Web-URLs	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)

Name	vertraut Standardeinstellung	Desktopbetriebssystem) Geltungsbereich
-------------	--	--

Load Management

Name	Standardeinstellung	Geltungsbereich
Toleranzwert für gleichzeitige Anmeldungen	2	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)
CPU-Nutzung	Deaktiviert	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)
CPU-Nutzung ausschließlich Prozesspriorität	Unter normal oder niedrig	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)
Datenträgenutzung	Deaktiviert	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)
Sitzungshöchstanzahl	250	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)
Speichernutzung	Deaktiviert	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)
Speichernutzung - Ausgangslast	Nulllast: 768 MB	Virtual Delivery Agent 7 (Serverbetriebssystem), 7.1 (Serverbetriebssystem), 7.5 (Serverbetriebssystem)

Profilverwaltung/Erweiterte Einstellungen

Name	Standardeinstellung	Geltungsbereich
Automatische Konfiguration deaktivieren	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Benutzer bei Problem abmelden	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Anzahl Wiederholungen beim Zugriff auf gesperrte Dateien	5	Alle Virtual Delivery Agent-Versionen

Internet-Cookiedateien bei Abmeldung verarbeiten	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Name	Standardeinstellung	Geltungsbereich

Profilverwaltung/Grundeinstellungen

Name	Standardeinstellung	Geltungsbereich
Aktiv zurückschreiben	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Profilverwaltung aktivieren	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Ausgeschlossene Gruppen	Deaktiviert: Mitglieder aller Benutzergruppen werden verarbeitet.	Alle Virtual Delivery Agent-Versionen
Unterstützung von Offlineprofilen	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Pfad zum Benutzerspeicher	Windows	Alle Virtual Delivery Agent-Versionen
Anmeldungen lokaler Administratoren verarbeiten	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Verarbeitete Gruppen	Deaktiviert: Mitglieder aller Benutzergruppen werden verarbeitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Plattformübergreifende Einstellungen

Name	Standardeinstellung	Geltungsbereich
Benutzergruppen für plattformübergreifende Einstellungen	Deaktiviert: Alle in der Richtlinieneinstellung Verarbeitete Gruppen angegebenen Benutzergruppen werden verarbeitet.	Alle Virtual Delivery Agent-Versionen
Plattformübergreifende Einstellungen aktivieren	Deaktiviert	Alle Virtual Delivery Agent-Versionen

Pfad zu plattformübergreifenden Definitionen	Deaktiviert: Kein Pfad angegeben.	Alle Virtual Delivery Agent-Versionen
Pfad zum Speicher für plattformübergreifende Einstellungen	Deaktiviert: Windows\PM_CM wird verwendet.	Alle Virtual Delivery Agent-Versionen
Quelle für plattformübergreifende Einstellungen	Deaktiviert	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Dateisystem/Ausschlüsse

Name	Standardeinstellung	Geltungsbereich
Ausschlussliste - Verzeichnisse	Deaktiviert: Alle Ordner im Benutzerprofil werden synchronisiert.	Alle Virtual Delivery Agent-Versionen
Ausschlussliste - Dateien	Deaktiviert: Alle Dateien im Benutzerprofil werden synchronisiert.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Dateisystem/Synchronisierung

Name	Standardeinstellung	Geltungsbereich
Zu synchronisierende Verzeichnisse	Deaktiviert: Nur nicht ausgeschlossene Ordner werden synchronisiert.	Alle Virtual Delivery Agent-Versionen
Zu synchronisierende Dateien	Deaktiviert: Nur nicht ausgeschlossene Dateien werden synchronisiert.	Alle Virtual Delivery Agent-Versionen
Zu spiegelnde Ordner	Deaktiviert: Es werden keine Ordner gespiegelt.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung

Name	Standardeinstellung	Geltungsbereich
Administratorzugriff gewähren	Deaktiviert	Alle Virtual Delivery Agent-Versionen

Name	Standardeinstellung	Geltungsbereich
Domenennamen einschließen	Deaktiviert	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung/AppData(Roaming)

Name	Standardeinstellung	Geltungsbereich
AppData(Roaming)-Pfad	Deaktiviert: Kein Speicherort angegeben.	Alle Virtual Delivery Agent-Versionen
Umleitungseinstellungen für AppData(Roaming)	Inhalte werden zu dem in der Richtlinieneinstellung AppData(Roaming)-Pfad angegebenen UNC-Pfad umgeleitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung/Kontakte

Name	Standardeinstellung	Geltungsbereich
'Kontakte'-Pfad	Deaktiviert: Kein Speicherort angegeben.	Alle Virtual Delivery Agent-Versionen
Umleitungseinstellungen für 'Kontakte'	Inhalte werden zu dem in der Richtlinieneinstellung 'Kontakte'-Pfad angegebenen UNC-Pfad umgeleitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung/Desktop

Name	Standardeinstellung	Geltungsbereich
'Desktop'-Pfad	Deaktiviert: Kein Speicherort angegeben.	Alle Virtual Delivery Agent-Versionen
Umleitungseinstellungen für 'Desktop'	Inhalte werden zu dem in der Richtlinieneinstellung 'Desktop'-Pfad angegebenen UNC-Pfad umgeleitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung/Dokumente

Name	Standardeinstellung	Geltungsbereich
'Dokumente'-Pfad	Deaktiviert: Kein Speicherort angegeben.	Alle Virtual Delivery Agent-Versionen

Name	Standardeinstellung	Geltungsbereich
Umleitungseinstellungen für 'Dokumente'	Inhalte werden zu dem in der Richtlinieneinstellung 'Dokumente'-Pfad angegebenen UNC-Pfad umgeleitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung/Downloads

Name	Standardeinstellung	Geltungsbereich
'Downloads'-Pfad	Deaktiviert: Kein Speicherort angegeben.	Alle Virtual Delivery Agent-Versionen
Umleitungseinstellungen für 'Downloads'	Inhalte werden zu dem in der Richtlinieneinstellung 'Downloads'-Pfad angegebenen UNC-Pfad umgeleitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung/Favoriten

Name	Standardeinstellung	Geltungsbereich
'Favoriten'-Pfad	Deaktiviert: Kein Speicherort angegeben.	Alle Virtual Delivery Agent-Versionen
Umleitungseinstellungen für 'Favoriten'	Inhalte werden zu dem in der Richtlinieneinstellung 'Favoriten'-Pfad angegebenen UNC-Pfad umgeleitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung/Links

Name	Standardeinstellung	Geltungsbereich
'Links'-Pfad	Deaktiviert: Kein Speicherort angegeben.	Alle Virtual Delivery Agent-Versionen
Umleitungseinstellungen für 'Links'	Inhalte werden zu dem in der Richtlinieneinstellung 'Links'-Pfad angegebenen UNC-Pfad umgeleitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung/Musik

Name	Standardeinstellung	Geltungsbereich
'Musik'-Pfad	Deaktiviert: Kein Speicherort angegeben.	Alle Virtual Delivery Agent-Versionen

Name	Standardeinstellung	Geltungsbereich
Umleitungseinstellungen für 'Musik'	Inhalte werden zu dem in der Richtlinieneinstellung 'Musik'-Pfad angegebenen UNC-Pfad umgeleitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung/Bilder

Name	Standardeinstellung	Geltungsbereich
'Bilder'-Pfad	Deaktiviert: Kein Speicherort angegeben.	Alle Virtual Delivery Agent-Versionen
Umleitungseinstellungen für 'Bilder'	Inhalte werden zu dem in der Richtlinieneinstellung 'Bilder'-Pfad angegebenen UNC-Pfad umgeleitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung/Gespeicherte Spiele

Name	Standardeinstellung	Geltungsbereich
'Gespeicherte Spiele'-Pfad	Deaktiviert: Kein Speicherort angegeben.	Alle Virtual Delivery Agent-Versionen
Umleitungseinstellungen für 'Gespeicherte Spiele'	Inhalte werden zu dem in der Richtlinieneinstellung 'Gespeicherte Spiele'-Pfad angegebenen UNC-Pfad umgeleitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung/Suchen

Name	Standardeinstellung	Geltungsbereich
'Suchen'-Pfad	Deaktiviert: Kein Speicherort angegeben.	Alle Virtual Delivery Agent-Versionen
Umleitungseinstellungen für 'Suchen'	Inhalte werden zu dem in der Richtlinieneinstellung 'Suchen'-Pfad angegebenen UNC-Pfad umgeleitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung/Startmenü

Name	Standardeinstellung	Geltungsbereich
Startmenü-Pfad	Deaktiviert: Kein Speicherort angegeben.	Alle Virtual Delivery Agent-Versionen

Name	Standardeinstellung	Geltungsbereich
Umleitungseinstellungen für 'Startmenü'	Inhalte werden zu dem in der Richtlinieneinstellung 'Startmenü'-Pfad angegebenen UNC-Pfad umgeleitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Ordnerumleitung/Videos

Name	Standardeinstellung	Geltungsbereich
'Videos'-Pfad	Deaktiviert: Kein Speicherort angegeben.	Alle Virtual Delivery Agent-Versionen
Umleitungseinstellungen für 'Videos'	Inhalte werden zu dem in der Richtlinieneinstellung 'Videos'-Pfad angegebenen UNC-Pfad umgeleitet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Protokolleinstellungen

Name	Standardeinstellung	Geltungsbereich
Active Directory-Aktionen	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Allgemeine Informationen	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Allgemeine Warnungen	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Protokollierung aktivieren	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Dateisystemaktionen	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Dateisystembenachrichtigungen	Deaktiviert	Alle Virtual Delivery Agent-Versionen

Abmeldung Name	Deaktiviert Standardeinstellung	Alle Virtual Geltungsbereich Delivery Agent- Versionen
Anmeldung	Deaktiviert	Alle Virtual Delivery Agent- Versionen
Max. Größe der Protokolldatei	1048576	Alle Virtual Delivery Agent- Versionen
Pfad zur Protokolldatei	Deaktiviert: Protokolldateien werden im Standardspeicherort gespeichert: %SystemRoot%\System32\Logfiles\UserProfileManager.	Alle Virtual Delivery Agent- Versionen
Persönliche Benutzerinformationen	Deaktiviert	Alle Virtual Delivery Agent- Versionen
Richtlinienwerte bei Anmeldung und Abmeldung	Deaktiviert	Alle Virtual Delivery Agent- Versionen
Registrierungsaktionen	Deaktiviert	Alle Virtual Delivery Agent- Versionen
Registrierungsunterschiede bei der Abmeldung	Deaktiviert	Alle Virtual Delivery Agent- Versionen

Profilverwaltung/Profilverarbeitung

Name	Standardeinstellung	Geltungsbereich
Verzögerung vor dem Löschen von zwischengespeicherten Profilen	0	Alle Virtual Delivery Agent- Versionen
Lokal zwischengespeicherte Profile nach Abmeldung löschen	Deaktiviert	Alle Virtual Delivery Agent-

Name	Standardeinstellung	Versionen Geltungsbereich
Behandlung von Konflikten lokaler Profile	Lokales Profil verwenden	Alle Virtual Delivery Agent-Versionen
Migration vorhandener Profile	Lokal und Roaming	Alle Virtual Delivery Agent-Versionen
Pfad zum Vorlagenprofil	Deaktiviert: Neue Benutzerprofile werden von dem Standardbenutzerprofil auf dem Gerät erstellt, auf dem sich ein Benutzer als Erstes anmeldet.	Alle Virtual Delivery Agent-Versionen
Vorlagenprofil überschreibt lokales Profil	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Vorlagenprofil überschreibt Roamingprofil	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Als verbindliches Citrix Profil für alle Anmeldungen verwendete Vorlagenprofil	Deaktiviert	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Registrierung

Name	Standardeinstellung	Geltungsbereich
Ausschlussliste	Deaktiviert: Alle Registrierungsschlüssel in der HKCU-Struktur werden verarbeitet, wenn ein Benutzer sich abmeldet.	Alle Virtual Delivery Agent-Versionen
Aufnahmeliste	Deaktiviert: Alle Registrierungsschlüssel in der HKCU-Struktur werden verarbeitet, wenn ein Benutzer sich abmeldet.	Alle Virtual Delivery Agent-Versionen

Profilverwaltung/Gestreamte Benutzerprofile

Name	Standardeinstellung	Geltungsbereich
Immer zwischenspeichern	Deaktiviert	Alle Virtual Delivery

Name	Standardeinstellung	Agent-Versionen Geltungsbereich
Immer Cachegröße	0 MBit	Alle Virtual Delivery Agent-Versionen
Profilstreaming	Deaktiviert	Alle Virtual Delivery Agent-Versionen
Gruppen mit gestreamten Benutzerprofilen	Deaktiviert: Alle Benutzerprofile in einer Organisationseinheit werden normal verarbeitet.	Alle Virtual Delivery Agent-Versionen
Timeout für bevorstehende Sperrdateien (Tage)	1 Tag	Alle Virtual Delivery Agent-Versionen

Receiver

Name	Standardeinstellung	Geltungsbereich
StoreFront-Kontenliste	Keine Stores angegeben	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)

Virtual Delivery Agent

Name	Standardeinstellung	Geltungsbereich
IPv6-Netzwerkmaske für Controllerregistrierung	Keine Netzwerkmaske angegeben.	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Controllerregistrierungsport	80	Alle Virtual Delivery Agent-Versionen
Controller-SIDs	Keine SIDs angegeben	Alle Virtual Delivery Agent-Versionen
Controller	Keine Controller angegeben	Alle Virtual Delivery Agent-Versionen
Automatische	Aktiviert	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem)

Name	Standardeinstellung	Geltungsbereich
Controllerupdates aktivieren		Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Nur IPv6-Controllerregistrierung verwenden	Deaktiviert	Virtual Delivery Agent 7 (Serverbetriebssystem und Desktopbetriebssystem), 7.1 (Serverbetriebssystem und Desktopbetriebssystem), 7.5 (Serverbetriebssystem und Desktopbetriebssystem)
Site-GUID	Kein GUID angegeben.	Alle Virtual Delivery Agent-Versionen

HDX 3D Pro

Name	Standardeinstellung	Geltungsbereich
Verlustfrei aktivieren	Aktiviert	Gilt für Virtual Delivery Agent 5.5 und 5.6 Feature Pack 1
HDX3DPro-Qualitätseinstellungen		Gilt für Virtual Delivery Agent 5.5 und 5.6 Feature Pack 1

Referenztablelle

Sep 29, 2015

Die folgenden Tabellen zeigen die Einstellungen, die Sie in einer Richtlinie konfigurieren können. In der linken Spalte finden Sie die Aufgaben, in der rechten die dazugehörigen Einstellungen.

Audio

Auszuführende Aufgabe:	Richtlinieneinstellung:
Steuern der Verwendung mehrerer Audiogeräte	Audio Plug & Play
Steuern, ob Audioeingaben vom Mikrofon auf dem Benutzergerät zulässig sind	Clientmikrofonumleitung
Steuern der Audioqualität auf dem Benutzergerät	Audioqualität
Steuern der Audiozuordnung für Lautsprecher am Benutzergerät	Clientaudioumleitung

Bandbreite für Benutzergeräte

Beschränken der Bandbreite	Richtlinieneinstellung:
Clientaudiozuordnung	• Bandbreitenlimit für die Audioumleitung oder • Bandbreitenlimit für die Audioumleitung (Prozent)
Kopieren und Einfügen mit der lokalen Zwischenablage	• Bandbreitenlimit für Zwischenablagenumleitung oder • Bandbreitenlimit für Zwischenablagenumleitung (Prozent)
Zugriff auf lokale Clientlaufwerke in einer Sitzung	• Bandbreitenlimit für Dateiumleitung oder • Bandbreitenlimit für Dateiumleitung (Prozent)
HDX MediaStream-Multimediabeschleunigung	• Bandbreitenlimit für HDX MediaStream-Multimediabeschleunigung oder • Bandbreitenlimit für HDX MediaStream-Multimediabeschleunigung (Prozent)
Clientsitzung	Bandbreitenlimit für Sitzung insgesamt

Drucken Beschränken der Bandbreite	Richtlinieneinstellung: - Bandbreitenlimit für Druckerumleitung oder - Bandbreitenlimit für Druckerumleitung (Prozent)
TWAIN-Geräte (wie Kameras oder Scanner)	- Bandbreitenlimit für TWAIN-Geräteumleitung oder - Bandbreitenlimit für TWAIN-Geräteumleitung (Prozent)
USB-Geräte	- Bandbreitenlimit für Client-USB-Geräteumleitung oder - Bandbreitenlimit für Client-USB-Geräteumleitung (Prozent)

Umleitung von Clientlaufwerken und Benutzergeräten

Auszuführende Aufgabe:	Richtlinieneinstellung:
Steuern, ob Laufwerke des Benutzergeräts verbunden werden, wenn Benutzer sich am Server anmelden	Clientlaufwerke automatisch verbinden
Steuern der Datenübertragung mit Kopier- und Einfügeoperationen zwischen dem Server und der lokalen Zwischenablage	Clientzwischenablagenumleitung
Steuern der Laufwerkzuordnung des Benutzergeräts	Clientlaufwerkumleitung
Steuern, ob die lokalen Festplatten des Benutzers in einer Sitzung verfügbar sind	- Lokale Clientlaufwerke und - Clientlaufwerkumleitung
Steuern, ob die lokalen Diskettenlaufwerke des Benutzers in einer Sitzung verfügbar sind	- Clientdiskettenlaufwerke und - Clientlaufwerkumleitung
Steuern, ob die Netzlaufwerke des Benutzers in einer Sitzung verfügbar sind	- Clientnetzlaufwerke und - Clientlaufwerkumleitung
Steuern, ob die lokalen CD-, DVD- oder Blu-ray-Laufwerke des Benutzers in einer Sitzung verfügbar sind	- Optische Clientlaufwerke und - Clientlaufwerkumleitung
Steuern, ob die lokalen Clientwechseldatenträger des Benutzers in einer Sitzung verfügbar sind	- Clientwechsellaufwerke und - Clientlaufwerkumleitung
Steuern, ob TWAIN-Geräte, wie Scanner und Kameras, in	TWAIN-Geräteumleitung für Client

Auszuführende Aufgabe: einer Sitzung verfügbar sind und Steuern der Komprimierung bei der Übertragung von Bilddaten	Richtlinieneinstellung: • TWAIN-Umleitung
Steuern, ob die USB-Geräte in einer Sitzung verfügbar sind	• Client-USB-Geräteumleitung und • Regeln für die Client-USB-Geräteumleitung
Geschwindigkeit beim Schreiben und Kopieren von Dateien auf einen Clientdatenträger über ein WAN erhöhen	Asynchrones Schreiben verwenden

Inhaltsumleitung

Auszuführende Aufgabe:	Richtlinieneinstellung:
Steuern der Verwendung der Inhaltsumleitung vom Server zum Benutzergerät	Host-zu-Client-Umleitung

Desktopbenutzeroberfläche

Auszuführende Aufgabe:	Richtlinieneinstellung:
Steuern, ob der Desktophintergrund in Benutzersitzungen angezeigt wird	Desktophintergrund
Anzeigen des Fensterinhalts beim Verschieben des Fensters	Fensterinhalt beim Verschieben anzeigen

Grafiken & Multimedia

Auszuführende Aufgabe:	Richtlinieneinstellung:
Steuern der maximalen Anzahl von Frames pro Sekunde, die an Benutzergeräte von virtuellen Desktops gesendet werden	Frameratesollwert
Steuern der visuellen Qualität der auf dem Benutzergerät angezeigten Bilder	Bildqualität
Steuern, ob Flash-Inhalte in Sitzungen wiedergegeben werden	Flash-Standardverhalten
Steuern, ob Flash-Inhalte auf Websites in Sitzungen angezeigt werden	• URL-Liste für serverseitigen Flash-Inhaltsabruf • Flash-URL-Kompatibilitätsliste

Auszuführende Aufgabe: Priorisieren des Multistream-Netzwerkdatenverkehrs	Richtlinieneinstellung:
Auszuführende Aufgabe:	Richtlinieneinstellung:
Angeben der Ports für ICA-Datenübertragungen über mehrere Verbindungen und Festlegen der Netzwerkprioritäten	Multiportrichtlinie
Aktivieren der Unterstützung von Multistreamverbindungen zwischen Servern und Benutzergeräten	Multistream (Computer- und Benutzereinstellungen)

Drucken

Auszuführende Aufgabe:	Richtlinieneinstellung:
Steuern der Clientdruckererstellung auf dem Benutzergerät	• Clientdrucker automatisch erstellen und • Clientdruckerumleitung
Steuern des Speicherorts für die Druckereigenschaften	Speicherung von Druckereigenschaften
Steuern, ob Druckanfragen vom Client oder vom Server verarbeitet werden	Direkte Verbindungen zu Druckservern
Steuern, ob Benutzer auf Drucker zugreifen können, die an die Benutzergeräte angeschlossen sind	Clientdruckerumleitung
Steuern, ob bei der automatischen Erstellung von Client- und Netzwerkdruckern native Windows-Treiber installiert werden	Automatische Installation von mitgelieferten Druckertreibern
Steuern, wann der universelle Druckertreiber verwendet wird	Verwendung universeller Druckertreiber
Wählen des Druckers anhand von Sitzungsinformationen eines mobilen Benutzers	Standarddrucker

Richtlinieneinstellungen für Connector für Configuration Manager 2012

Sep 29, 2015

Der Abschnitt "Connector für Configuration Manager 2012" enthält Richtlinieneinstellungen zum Konfigurieren des Citrix Connector 7.5-Agents.

Wichtig: Richtlinien für Warnungs-, Abmeldungs- und Neustartmeldungen gelten nur für Bereitstellungen für Serverbetriebssystemmaschinenkataloge, die manuell oder über Provisioning Services verwaltet werden. Bei solchen Maschinenkatalogen benachrichtigt der Connector-Dienst Benutzer über ausstehende Anwendungsinstallationen oder Softwareupdates.

Verwenden Sie bei über MCS verwalteten Katalogen Studio zur Benachrichtigung der Benutzer. Verwenden Sie bei manuell verwalteten Maschinenkatalogen mit Desktopbetriebssystemmaschinen Configuration Manager zur Benachrichtigung der Benutzer. Verwenden Sie bei mit Provisioning Services verwalteten Maschinenkatalogen mit Desktopbetriebssystemmaschinen Provisioning Services zur Benachrichtigung der Benutzer.

Zeitraum für Vorabwarnung

Diese Einstellung gibt das Intervall an, mit dem Benutzern Vorabmeldungen angezeigt werden.

Intervalle werden im Format ttt.hh:mm:ss festgelegt. Dabei gilt Folgendes:

- ttt steht für die Tage. Dieser Parameter ist optional und kann Werte von 0 bis 999 annehmen.
- hh steht für die Stunden und kann Werte von 0 bis 23 annehmen.
- mm steht für Minuten und kann Werte von 0 bis 59 annehmen.
- ss steht für Sekunden und kann Werte von 0 bis 59 annehmen.

Das Standardintervall ist 1 Stunde (01:00:00).

Meldungsfeldtext für Vorabwarnung

Diese Einstellung enthält den editierbaren Text für die Vorabmeldung, die Benutzer vor anstehenden Softwareupdates oder Wartungsaufgaben erhalten, für die sie sich abmelden müssen.

Die Standardmeldung lautet: {TIMESTAMP} Please save your work. The server will go offline for maintenance in {TIMELEFT}

Meldungsfeldtitel für Vorabwarnung

Diese Einstellung enthält den editierbaren Titel für die Titelleiste der Vorabmeldung, die Benutzer erhalten.

Der Standardtitel ist: Upcoming Maintenance

Zeitraum für Vorabwarnung

Diese Einstellung definiert, wie lange vor Wartungsaufgaben die Vorabwarnmeldung zum ersten Mal angezeigt wird.

Der Zeitraum wird im Format ttt.hh:mm:ss angegeben, mit den folgenden Variablen:

- ttt steht für die Tage. Dieser Parameter ist optional und kann Werte von 0 bis 999 annehmen.
- hh steht für die Stunden und kann Werte von 0 bis 23 annehmen.
- mm steht für Minuten und kann Werte von 0 bis 59 annehmen.

- ss steht für Sekunden und kann Werte von 0 bis 59 annehmen.

In der Standardeinstellung ist der Wert 16 Stunden (16:00:00), d. h. dass die erste Vorabwarnmeldung ca. 16 Stunden vor der Wartung angezeigt wird.

Letzter Meldungsfeldtext für Abmeldung erzwingen

Diese Einstellung enthält den editierbaren Text für die Meldung, die Benutzer warnt, dass die Abmeldung erzwungen wird.

Die Standardmeldung lautet: The server is currently going offline for maintenance

Letzter Meldungsfeldtitel für Abmeldung erzwingen

Diese Einstellung enthält den editierbaren Titel für die Titelleiste der letzten Meldung für Abmeldung erzwingen.

Der Standardtitel lautet: Notification From IT Staff

Kulanzzeitraum für erzwungenes Abmelden

Diese Einstellung definiert den Kulanzzeitraum, der Benutzern zugestanden wird, nachdem sie gewarnt wurden, dass die Abmeldung erzwungen wird, und dem tatsächlichen erzwungenen Abmelden, damit die ausstehenden Wartungsaufgaben gestartet werden können.

Der Zeitraum wird im Format ttt.hh:mm:ss angegeben, mit den folgenden Variablen:

- ttt steht für die Tage. Dieser Parameter ist optional und kann Werte von 0 bis 999 annehmen.
- hh steht für die Stunden und kann Werte von 0 bis 23 annehmen.
- mm steht für Minuten und kann Werte von 0 bis 59 annehmen.
- ss steht für Sekunden und kann Werte von 0 bis 59 annehmen.

In der Standardeinstellung ist der Kulanzzeitraum für erzwungenes Abmelden auf 5 Minuten (00:05:00) festgelegt.

Letzter Meldungsfeldtext für Abmeldung erzwingen

Diese Einstellung enthält den editierbaren Text für die letzte Warnmeldung, die Benutzer auffordert, ihre Arbeit zu speichern und sich vor der erzwungenen Abmeldung abzumelden.

Die Standardmeldung enthält den folgenden Text: "{TIMESTAMP} Please save your work and log off. The server will go offline for maintenance in {TIMELEFT}"

Meldungsfeldtitel für Abmeldung erzwingen

Diese Einstellung enthält den editierbaren Text für die Titelleiste der Meldung für Abmeldung erzwingen.

Der Standardtitel lautet: Notification From IT Staff

Imageverwalteter Modus

Der Connector-Agent erkennt automatisch, wenn er auf einem von Provisioning Services oder MCS verwalteten Maschinenklon ausgeführt wird. Der Agent blockiert Configuration Manager-Updates auf imageverwalteten Klonen und installiert die Updates automatisch auf dem Masterimage des Katalogs.

Nachdem ein Masterimage aktualisiert wurde, verwenden Sie Studio zum Orchestrieren des Neustarts der MCS-Klone. Der Connector-Agent orchestriert automatisch den Neustart von PVS-Katalogklonen während der Configuration Manager-Wartung. Zur Außerkraftsetzung dieses Verhaltens, damit Software auf Katalogklonen von Configuration Manager

installiert wird, ändern Sie den Modus von "Imageverwaltet" in Deaktiviert.

Meldungsfeldtext für Neustarten

Diese Einstellung enthält den editierbaren Text der Benutzermeldung, dass der Server bald neu gestartet wird.

Die Standardmeldung lautet: The server is currently going offline for maintenance

Normales Zeitintervall, in dem die Agent-Aufgabe ausgeführt wird

Durch diese Einstellung wird festgelegt, wie häufig der Citrix Connector Agent-Aufgabe ausgeführt wird.

Der Zeitraum wird im Format ttt.hh:mm:ss angegeben, mit den folgenden Variablen:

- ttt steht für die Tage. Dieser Parameter ist optional und kann Werte von 0 bis 999 annehmen.
- hh steht für die Stunden und kann Werte von 0 bis 23 annehmen.
- mm steht für Minuten und kann Werte von 0 bis 59 annehmen.
- ss steht für Sekunden und kann Werte von 0 bis 59 annehmen.

In der Standardeinstellung ist das Intervall auf fünf Minuten (00:05:00) festgelegt.

Einstellungen der Richtlinie "ICA"

Sep 29, 2015

Der Abschnitt "ICA" enthält Richtlinieneinstellungen für ICA-Listenerverbindungen und die Zwischenablagenzuordnung.

Clientzwischenablagenumleitung

Mit dieser Einstellung legen Sie fest, ob die Zwischenablage auf dem Clientgerät der Zwischenablage auf dem Server zugeordnet wird.

Standardmäßig ist die Umleitung der Zwischenablage zugelassen.

Wenn Sie verhindern möchten, dass Daten durch Kopieren und Einfügen über die Zwischenablage zwischen einer Sitzung und der lokalen Zwischenablage übertragen werden, wählen Sie Nicht zugelassen. Benutzer können weiterhin die Zwischenablage für das Kopieren von Daten zwischen Anwendungen einsetzen, die in Sitzungen ausgeführt werden.

Nachdem diese Einstellung zugelassen wurde, konfigurieren Sie die maximal zulässige Bandbreite, die die Zwischenablage in einer Clientverbindung verbrauchen darf. Verwenden Sie dazu die Einstellung Bandbreitenlimit für Zwischenablagenumleitung oder Bandbreitenlimit für Zwischenablagenumleitung (Prozent).

Desktop starten

Mit dieser Einstellung legen Sie fest, ob Nicht-Administratoren eine Verbindung zu einer Desktopsitzung auf dem Server herstellen können.

Standardmäßig können Benutzer ohne Administratorrechte keine Verbindung zu diesen Sitzungen herstellen.

ICA-Listener - Verbindungstimeout

Mit dieser Einstellung geben Sie die maximale Wartezeit an, bis eine Verbindung mit dem ICA-Protokoll abgeschlossen wird.

Standardmäßig ist die maximale Wartezeit 120000 Millisekunden oder zwei Minuten.

ICA-Listenerportnummer

Mit dieser Einstellung konfigurieren Sie die TCP/IP-Portnummer, die vom ICA-Protokoll auf dem Server verwendet wird.

Die Standardeinstellung der Portnummer ist "1494".

Gültige Portnummern müssen zwischen 0 und 65535 liegen. Sie dürfen keinen Konflikt mit anderen gängigen Portnummern verursachen. Wenn Sie die Portnummer ändern, muss der Server neu gestartet werden, damit der neue Wert wirksam werden kann. Wenn Sie die Portnummer auf dem Server ändern, müssen Sie sie auch in jedem Receiver oder Plug-In ändern, das eine Verbindung zu diesem Server herstellt.

Starten nicht-veröffentlichter Programme bei Clientverbindung

Mit dieser Einstellung geben Sie an, ob Startanwendungen oder veröffentlichte Anwendungen über ICA oder RDP auf dem Server gestartet werden.

Standardmäßig dürfen nur veröffentlichte Anwendungen gestartet werden.

Lesezugriff auf Zwischenablage

Mit dieser Einstellung aktivieren oder deaktivieren Sie das Kopieren und Einfügen von Daten aus der Sitzung auf dem lokalen Benutzergerät.

Diese Einstellung ist standardmäßig deaktiviert und Benutzer können Daten zwischen der Sitzung und dem lokalen Benutzergerät kopieren und einfügen.

Wenn die Option aktiviert ist, können Benutzer nur Daten von dem lokalen Benutzergerät ausschneiden und in die Sitzung einfügen.

Wenn Sie diese Einstellung aktivieren, muss auch für die Einstellung Clientzwischenablagenumleitung die Option Zugelassen gewählt werden. Wenn diese Einstellung deaktiviert ist, wird die Zwischenablage auf dem Benutzergerät nicht der Zwischenablage auf dem Server zugeordnet und Benutzer können keine Daten zwischen Sitzung und lokalem Benutzergerät kopieren und einfügen.

Einstellungen der Richtlinie "Automatische Wiederverbindung von Clients"

Der Abschnitt "Automatische Wiederverbindung von Clients" enthält Richtlinieneinstellungen, mit denen Sie die automatische Wiederverbindung von Sitzungen steuern.

Automatische Wiederverbindung von Clients

Diese Einstellung legt fest, ob die automatische Wiederverbindung des gleichen Clients zulässig ist, nachdem eine Verbindung unterbrochen wurde.

Standardmäßig ist die automatische Wiederverbindung zugelassen.

Ist die automatische Wiederverbindung zulässig, können Benutzer ihre Arbeit an der Stelle wiederaufnehmen, an der die Verbindung unterbrochen wurde. Die automatische Wiederverbindung erkennt unterbrochene Verbindungen und verbindet die Benutzer wieder mit ihren Sitzungen.

Bei der automatischen Wiederverbindung kann es aber dazu kommen, dass eine neue Sitzung gestartet wird (statt die Verbindung mit der bestehenden Sitzung wiederherzustellen), wenn das Receiver-Cookie, das den Schlüssel für Sitzungs-ID und Anmeldeinformationen enthält, nicht verwendet wird. Das Cookie wird nicht verwendet, wenn es abgelaufen ist, z. B. weil die Wiederverbindung verzögert wird, oder wenn die Anmeldeinformationen neu eingegeben werden müssen. Die automatische Wiederverbindung von Clients wird nicht ausgelöst, wenn Benutzer die Sitzung absichtlich trennen.

Authentifizierung bei automatischer Wiederverbindung von Clients

Mit dieser Einstellung ist die Authentifizierung erforderlich, wenn die Verbindung zum Client automatisch wiederhergestellt wird.

Standardmäßig ist die Authentifizierung nicht erforderlich.

Wenn sich ein Benutzer erstmals anmeldet, werden seine Anmeldeinformationen verschlüsselt und gespeichert und es wird ein Cookie mit dem Schlüssel erstellt, der an Receiver gesendet wird. Wenn diese Einstellung konfiguriert ist, werden keine Cookies verwendet. Stattdessen wird ein Dialogfeld mit der Aufforderung zur Eingabe der Anmeldeinformationen angezeigt, wenn Receiver versucht, die Verbindung automatisch wiederherzustellen.

Protokollierung der automatischen Wiederverbindung von Clients

Mit dieser Einstellung legen Sie fest, ob die automatischen Wiederverbindungen im Ereignisprotokoll aufgezeichnet werden.

Standardmäßig ist die Protokollierung deaktiviert.

Wenn die Protokollierung aktiviert ist, werden Informationen über erfolgreiche und fehlgeschlagene Wiederverbindungsereignisse im Systemprotokoll des Servers aufgezeichnet. Eine Site stellt kein kombiniertes Protokoll zu Wiederverbindungsereignissen auf allen Servern zur Verfügung.

Einstellungen der Richtlinie "Audio"

Der Abschnitt "Audio" enthält Richtlinieneinstellungen, mit denen Sie das Senden und Empfangen von Audiodaten auf dem Benutzergerät konfigurieren können, ohne dass es dabei zu einer unerwünschten Verschlechterung der Leistung kommt.

Audio über UDP - Real-time Transport

Diese Einstellung aktiviert bzw. deaktiviert die Audioübertragung und den Audioempfang zwischen VDA und Benutzergeräten über RTP mit UDP (User Datagram Protocol). Wenn diese Einstellung deaktiviert ist, wird Audio über TCP gesendet und empfangen.

Standardmäßig ist Audio über UDP zugelassen.

Audio Plug & Play

Mit dieser Einstellung lassen Sie die Verwendung mehrerer Audiogeräte zum Aufzeichnen und zum Wiedergeben von Ton zu oder verhindern sie.

Standardmäßig ist die Verwendung mehrerer Audiogeräte zulässig.

Audioqualität

Mit dieser Einstellung legen Sie die Tonqualität fest, die in Benutzersitzungen empfangen wird.

In der Standardeinstellung ist die Tonqualität auf Hoch - High Definition-Audio eingestellt.

Um die Tonqualität zu steuern, wählen Sie eine der folgenden Optionen:

- Wählen Sie Gering - für langsame Verbindungen für Verbindungen mit geringer Bandbreite. An den Client gesendete Audiodaten werden bis auf 16 Kbit/s komprimiert. Diese Komprimierung führt zu einer erheblichen Verringerung der Tonqualität, ermöglicht aber eine akzeptable Leistung bei einer Verbindung mit geringer Bandbreite.
- Wählen Sie Mittel - optimiert für Sprache für die meisten LAN-basierten Verbindungen. An den Client gesendete Audiodaten werden bis auf 64 Kbit/s komprimiert. Dieses Codec bietet eine schnelle Codierung und ist daher ideal für Softphones und Unified Communications-Anwendungen geeignet, wenn Sie eine serverseitige Medienverarbeitung benötigen.
- Wählen Sie Hoch - High Definition Audio für Verbindungen, bei denen die Bandbreite keine Rolle spielt und bei denen die Tonqualität wichtig ist. Clients können Audiodaten mit der nativen Abspielrate wiedergeben. Audiodaten können bis zu 1,3 MBit/s Bandbreite verbrauchen, um klar wiedergegeben zu werden. Die Übertragung dieser Datenmenge kann zu einer höheren CPU-Belastung und Engpässen im Netzwerk führen.

Die Bandbreite wird nur verbraucht, während Audio aufgenommen oder abgespielt wird. Wenn beides gleichzeitig stattfindet, verdoppelt sich der Bandbreitenverbrauch.

Konfigurieren Sie die Einstellungen Bandbreitenlimit für die Audioumleitung oder Bandbreitenlimit für die Audioumleitung (Prozent), um die maximale Bandbreite anzugeben.

Clientaudioumleitung

Mit dieser Einstellung legen Sie fest, ob auf dem Server gehostete Anwendungen Audiodateien über ein auf dem Benutzergerät installiertes Audiogerät wiedergeben können. Diese Einstellung gibt auch an, ob Benutzer Audio aufzeichnen können.

Standardmäßig ist die Audioumleitung zugelassen.

Nachdem Sie diese Einstellung zugelassen haben, können Sie die Bandbreite beschränken, die durch die Wiedergabe oder das Aufzeichnen von Audio verbraucht wird. Durch Beschränken der Bandbreite, die durch Audio verbraucht wird, kann sich die Anwendungsleistung steigern, die Audioqualität wird aber herabgesetzt. Die Bandbreite wird nur verbraucht, während Audio aufgenommen oder abgespielt wird. Wenn beides gleichzeitig stattfindet, verdoppelt sich der Bandbreitenverbrauch.

Konfigurieren Sie die Einstellungen Bandbreitenlimit für die Audioumleitung oder Bandbreitenlimit für die Audioumleitung (Prozent), um die maximale Bandbreite anzugeben.

Clientmikrofonumleitung

Mit dieser Einstellung aktivieren oder deaktivieren Sie die Umleitung von Clientmikrofonen. Wenn aktiviert, können Benutzer Mikrofone für die Aufnahme von Audioeingaben in einer Sitzung verwenden.

Standardmäßig ist die Clientmikrofonumleitung zugelassen.

Aus Sicherheitsgründen werden Benutzer darauf hingewiesen, wenn Server, die keine vertrauenswürdige Beziehung zu den Geräten haben, auf Mikrofone zugreifen. Benutzer können den Zugriff ermöglichen oder ablehnen. Benutzer können die Warnung in Citrix Receiver deaktivieren.

Wenn die Einstellung Clientaudioumleitung auf dem Benutzergerät deaktiviert ist, hat diese Regel keine Auswirkung.

Einstellungen der Richtlinie "Bandbreite"

Aktualisiert: 2014-02-27

Der Abschnitt "Bandbreite" enthält Richtlinieneinstellungen, mit denen Sie Leistungsprobleme vermeiden können, die sich aus der Bandbreitenverwendung in der Clientsitzung ergeben.

Wichtig: Die Verwendung dieser Richtlinieneinstellungen mit den Richtlinieneinstellungen Multistream kann zu unerwarteten Ergebnissen führen. Wenn Sie Multistreameinstellungen in einer Richtlinie verwenden, stellen Sie sicher, dass diese Richtlinieneinstellungen für das Bandbreitenlimit nicht eingeschlossen sind.

Bandbreitenlimit für die Audioumleitung

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite für die Wiedergabe oder die Aufnahme von Audio in einer Benutzersitzung in Kilobits pro Sekunde an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für die Audioumleitung (Prozent) einen Wert angeben, wird die restriktivere Einstellung angewendet.

Bandbreitenlimit für die Audioumleitung (Prozent)

Mit dieser Einstellung geben Sie das maximal zulässige Bandbreitenlimit für die Wiedergabe oder die Aufnahme von Audio in

als Prozentsatz der Gesamtsitzungsbandbreite an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für die Audioumleitung einen Wert angeben, wird die restriktivere Einstellung angewendet.

Wenn Sie die Einstellung konfigurieren, müssen Sie auch die Einstellung Bandbreitenlimit für Sitzung insgesamt konfigurieren, mit der die Gesamtbandbreite definiert wird, die für Clientsitzungen verfügbar ist.

Bandbreitenlimit für Client-USB-Geräteumleitung

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite für die Umleitung von USB-Geräten zum und vom Client in Kilobit pro Sekunde an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für Client-USB-Geräteumleitung (Prozent) einen Wert angeben, wird die restriktivere Einstellung angewendet.

Bandbreitenlimit für Client-USB-Geräteumleitung (Prozent)

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite für die Umleitung von USB-Geräten zum und vom Client als Prozentsatz der Gesamtsitzungsbandbreite an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für Client-USB-Geräteumleitung einen Wert angeben, wird die restriktivere Einstellung angewendet.

Wenn Sie die Einstellung konfigurieren, müssen Sie auch die Einstellung Bandbreitenlimit für Sitzung insgesamt konfigurieren, mit der die Gesamtbandbreite definiert wird, die für Clientsitzungen verfügbar ist.

Bandbreitenlimit für Zwischenablagenumleitung

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite in Kilobits pro Sekunde für Datenübertragungen zwischen einer Sitzung und der lokalen Zwischenablage an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für Zwischenablagenumleitung (Prozent) einen Wert angeben, wird die restriktivere Einstellung angewendet.

Bandbreitenlimit für Zwischenablagenumleitung (Prozent)

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite für Datenübertragungen zwischen einer Sitzung und der lokalen Zwischenablage als Prozentsatz der Gesamtsitzungsbandbreite an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für Zwischenablagenumleitung einen Wert angeben, wird die restriktivere Einstellung angewendet.

Wenn Sie die Einstellung konfigurieren, müssen Sie auch die Einstellung Bandbreitenlimit für Sitzung insgesamt konfigurieren, mit der die Gesamtbandbreite definiert wird, die für Clientsitzungen verfügbar ist.

Bandbreitenlimit für COM-Portumleitung

Hinweis: Konfigurieren Sie diese Einstellung für Virtual Delivery Agent 7.x über die Registrierung. Weitere Informationen finden Sie unter [Konfigurieren von COM-Port- und LPT-Portumleitungseinstellungen in der Registrierung](#).

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite für den Zugriff auf einen COM-Port in einer Clientverbindung in Kilobits pro Sekunde an. Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für COM-Portumleitung (Prozent) einen Wert angeben, wird die restriktivere Einstellung angewendet.

Bandbreitenlimit für COM-Portumleitung (Prozent)

Hinweis: Konfigurieren Sie diese Einstellung für Virtual Delivery Agent 7.x über die Registrierung. Weitere Informationen finden Sie unter [Konfigurieren von COM-Port- und LPT-Portumleitungseinstellungen in der Registrierung](#).

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite für den Zugriff auf COM-Ports in einer Clientverbindung als Prozentsatz der Gesamtsitzungsbandbreite an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für COM-Portumleitung einen Wert angeben, wird die restriktivere Einstellung angewendet.

Wenn Sie die Einstellung konfigurieren, müssen Sie auch die Einstellung Bandbreitenlimit für Sitzung insgesamt konfigurieren, mit der die Gesamtbandbreite definiert wird, die für Clientsitzungen verfügbar ist.

Bandbreitenlimit für Dateiumleitung

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite für den Zugriff auf Clientlaufwerke in einer Clientverbindung in Kilobits pro Sekunde an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für Dateiumleitung (Prozent) einen Wert angeben, wird die restriktivere Einstellung (mit dem niedrigeren Wert) angewendet.

Bandbreitenlimit für Dateiumleitung (Prozent)

Mit dieser Einstellung geben Sie das maximal zulässige Bandbreitenlimit für den Zugriff auf Clientlaufwerke als Prozentsatz der Gesamtsitzungsbandbreite an

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für Dateiumleitung einen Wert angeben, wird die restriktivere Einstellung angewendet.

Wenn Sie die Einstellung konfigurieren, müssen Sie auch die Einstellung Bandbreitenlimit für Sitzung insgesamt konfigurieren, mit der die Gesamtbandbreite definiert wird, die für Clientsitzungen verfügbar ist.

Bandbreitenlimit für HDX MediaStream-Multimediabeschleunigung

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite für die Bereitstellung von Streamingaudio und -video mit HDX MediaStream-Multimediabeschleunigung in Kilobit pro Sekunde an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für HDX MediaStream-Multimediabeschleunigung (Prozent) einen Wert angeben, wird die restriktivere Einstellung angewendet.

Bandbreitenlimit für HDX MediaStream-Multimediabeschleunigung (Prozent)

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite für die Bereitstellung von Streamingaudio und -video mit HDX MediaStream-Multimediabeschleunigung als Prozentsatz der Gesamtsitzungsbandbreite an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung einen Wert angeben und auch für die Einstellung "Bandbreitenlimit für HDX MediaStream-Multimediabeschleunigung", wird die restriktivere Einstellung angewendet.

Wenn Sie die Einstellung konfigurieren, müssen Sie auch die Einstellung Bandbreitenlimit für Sitzung insgesamt konfigurieren, mit der die Gesamtbandbreite definiert wird, die für Clientsitzungen verfügbar ist.

Bandbreitenlimit für LPT-Portumleitung

Hinweis: Konfigurieren Sie diese Einstellung für Virtual Delivery Agent 7.x über die Registrierung. Weitere Informationen finden Sie unter [Konfigurieren von COM-Port- und LPT-Portumleitungseinstellungen in der Registrierung](#).

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite in Kilobits pro Sekunde an, die für Druckaufträge über den LPT-Port in einer Benutzersitzung verwendet werden kann.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für LPT-Portumleitung (Prozent) einen Wert angeben, wird die restriktivere Einstellung angewendet.

Bandbreitenlimit für LPT-Portumleitung (Prozent)

Hinweis: Konfigurieren Sie diese Einstellung für Virtual Delivery Agent 7.x über die Registrierung. Weitere Informationen finden Sie unter [Konfigurieren von COM-Port- und LPT-Portumleitungseinstellungen in der Registrierung](#).

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite, die für Druckaufträge über den LPT-Port in einer Sitzung verwendet werden darf, als Prozent der Gesamtsitzungsbandbreite an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und auch für die Einstellung Bandbreitenlimit für LPT-Portumleitung einen Wert angeben, wird die restriktivere Einstellung angewendet.

Wenn Sie die Einstellung konfigurieren, müssen Sie auch die Einstellung Bandbreitenlimit für Sitzung insgesamt konfigurieren, mit der die Gesamtbandbreite definiert wird, die für Clientsitzungen verfügbar ist.

Bandbreitenlimit für Sitzung insgesamt

Mit dieser Einstellung geben Sie Gesamtbandbreite in Kilobits pro Sekunde an, die für Benutzersitzungen verwendet werden

kann.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Durch Beschränken der Bandbreite, die von einer Clientverbindung verbraucht wird, kann zu einer Leistungsverbesserung führen, wenn andere Anwendungen außerhalb der Clientverbindung auch auf die Bandbreite zugreifen.

Bandbreitenlimit für Druckerumleitung

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite für den Zugriff auf Clientdrucker in einer Benutzersitzung in Kilobits pro Sekunde an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für Druckerumleitung (Prozent) einen Wert angeben, wird die restriktivere Einstellung angewendet.

Bandbreitenlimit für Druckerumleitung (Prozent)

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite für den Zugriff auf Clientdrucker als Prozentsatz der Gesamtsitzungsbandbreite an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für Druckerumleitung einen Wert angeben, wird die restriktivere Einstellung angewendet.

Wenn Sie die Einstellung konfigurieren, müssen Sie auch die Einstellung Bandbreitenlimit für Sitzung insgesamt konfigurieren, mit der die Gesamtbandbreite definiert wird, die für Clientsitzungen verfügbar ist.

Bandbreitenlimit für TWAIN-Geräteumleitung

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite in Kilobits pro Sekunde für die Steuerung von TWAIN-Bildverarbeitungsgeräten in veröffentlichten Anwendungen an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für TWAIN-Geräteumleitung (Prozent) einen Wert angeben, wird die restriktivere Einstellung angewendet.

Bandbreitenlimit für TWAIN-Geräteumleitung (Prozent)

Mit dieser Einstellung geben Sie die maximal zulässige Bandbreite für die Steuerung von TWAIN-Bildverarbeitungsgeräten in veröffentlichten Anwendungen als Prozentsatz der Gesamtsitzungsbandbreite an.

Standardmäßig ist kein Maximalwert (Null) angegeben.

Wenn Sie für diese Einstellung und für die Einstellung Bandbreitenlimit für TWAIN-Geräteumleitung einen Wert angeben, wird die restriktivere Einstellung angewendet.

Wenn Sie die Einstellung konfigurieren, müssen Sie auch die Einstellung Bandbreitenlimit für Sitzung insgesamt konfigurieren, mit der die Gesamtbandbreite definiert wird, die für Clientsitzungen verfügbar ist.

Einstellungen der Richtlinie "Clientsensoren"

Der Abschnitt "Clientsensoren" enthält Richtlinieneinstellungen, mit denen gesteuert wird, wie Informationen über den Mobilgerätsensor in einer Benutzersitzung gehandhabt werden.

Anwendungen können den physischen Standort des Clientgeräts verwenden

Diese Einstellung legt fest, ob Anwendungen, die in einer Sitzung auf einem Mobilgerät ausgeführt werden, den physischen Standort des Benutzergeräts verwenden können.

In der Standardeinstellung ist die Verwendung von Standortinformationen nicht zugelassen.

Wenn diese Einstellung nicht zugelassen ist und eine Anwendung versucht, die Standortinformationen abzurufen, wird ein Wert von "Zugriff verweigert" zurückgegeben.

Wenn diese Einstellung nicht zugelassen ist, kann ein Benutzer die Verwendung von Standortinformationen verhindern und eine Receiver-Anforderung für den Zugriff auf den Standort ablehnen. Android- und iOS-Geräte senden am Anfang jeder Sitzung eine Anforderung für die Standortinformationen.

Berücksichtigen Sie beim Entwickeln von gehosteten Anwendungen, die die Einstellung Anwendungen können den physischen Standort des Clientgeräts verwenden enthalten Folgendes:

- Eine standortaktivierte Anwendung sollte sich aus den folgenden Gründen nicht darauf verlassen, dass Standortinformationen verfügbar sind:
 - Ein Benutzer gewährt möglicherweise keinen Zugriff auf die Standortinformationen.
 - Der Standort ist ggf. nicht verfügbar oder ändert sich, während die Anwendung ausgeführt wird.
 - Ein Benutzer stellt möglicherweise eine Verbindung mit der Anwendungssitzung von einem anderen Gerät her, das keine Standortinformationen unterstützt.
- Anforderungen für eine standortaktivierte Anwendung:
 - Das Standortfeature muss in der Standardeinstellung deaktiviert sein.
 - Eine Benutzeroption für das Zulassen oder Ablehnen des Features muss bei Ausführung der Anwendung verfügbar sein.
 - Eine Benutzeroption muss verfügbar sein, mit der von der Anwendung zwischengespeicherte Standortdaten gelöscht werden. (Receiver speichert keine Standortdaten im Cache.)
- Eine standortaktivierte Anwendung muss die Granularität der Standortinformationen verwalten, damit die abgefragten Daten dem Zweck der Anwendung entsprechen und die entsprechenden Gesetze einhalten.
- Bei der Verwendung der Standortdienste sollte eine sichere Verbindung (zum Beispiel mit SSL/TLS oder einem VPN) erzwungen werden. Citrix Receiver sollte eine Verbindung mit vertrauenswürdigen Servern herstellen.
- Sie sollten eine Rechtsberatung hinsichtlich der Verwendung von Standortdiensten erwägen.

Einstellungen der Richtlinie "Desktopbenutzeroberfläche"

Aktualisiert: 2014-01-27

Der Abschnitt "Desktopbenutzeroberfläche" enthält Richtlinieneinstellungen für visuelle Effekte, wie Desktophintergrund, Menüanimationen und das Verhalten von Fensterinhalten beim Drag & Drop, um die für Clientverbindungen verbrauchte Bandbreite zu steuern. Die Anwendungsleistung über ein WAN lässt sich durch Beschränken des Bandbreitenverbrauchs verbessern.

Desktopgestaltungsumleitung

Mit dieser Einstellung geben Sie an, ob die Verarbeitung des Grafikprozessors (GPU) oder des integrierten Grafikprozessors (IGP) auf dem Benutzergerät für die lokale DirectX-Grafikwiedergabe verwendet werden soll, um eine nahtlosere Windows-Desktopdarstellung zu erzielen. Wenn die Desktopgestaltungsumleitung aktiviert ist, wird eine hoch reaktionsfähige Windows-Benutzererfahrung bei Beibehaltung einer hohen Skalierbarkeit auf dem Server gewährleistet.

Standardmäßig ist die Desktopgestaltungsumleitung aktiviert.

Um die Desktopgestaltungsumleitung auszuschalten und die für Benutzersitzungen erforderliche Bandbreite zu reduzieren, wählen Sie Deaktiviert aus, wenn Sie diese Einstellung einer Richtlinie hinzufügen.

Grafikqualität Desktopgestaltung

Mit dieser Einstellung wird die Qualität der für die Desktopgestaltungsumleitung verwendeten Grafiken angegeben.

Die Standardeinstellung ist "Hoch".

Wählen Sie die Qualität Hoch, Mittel, Niedrig oder Verlustfrei aus.

Desktophintergrund

Mit dieser Einstellung legen Sie fest, ob Hintergründe in Benutzersitzungen angezeigt werden.

Standardmäßig kann der Desktophintergrund in Benutzersitzungen angezeigt werden.

Um den Desktophintergrund auszuschalten und die für Benutzersitzungen erforderliche Bandbreite zu reduzieren, wählen Sie die Einstellung Nicht zugelassen, wenn Sie diese Einstellung einer Richtlinie hinzufügen.

Menüanimation

Mit dieser Einstellung legen Sie fest, ob Menüanimation in Benutzersitzungen zugelassen oder verhindert wird.

Standardmäßig ist Menüanimation zugelassen.

Menüanimationen sind eine Microsoft-Einstellung, bei der Menüs nach einer kurzen Verzögerung durch Bildlauf- oder Einblendeeffekt angezeigt werden. Wenn für diese Richtlinieneinstellung Zugelassen festgelegt wurde, wird unten im Menü ein Pfeil angezeigt. Das Menü wird eingeblendet, wenn Sie mit der Maus auf diesen Pfeil zeigen.

Fensterinhalt beim Verschieben anzeigen

Mit dieser Einstellung legen Sie fest, ob Fensterinhalte beim Verschieben des Fensters auf dem Bildschirm angezeigt werden.

Standardmäßig ist die Anzeige des Fensterinhalts beim Verschieben zugelassen.

Wenn Zugelassen ausgewählt ist, wird beim Verschieben das ganze Fenster angezeigt. Wenn Nicht zugelassen ausgewählt ist, wird bis zum Ablegen nur der Fensterrahmen beim Verschieben angezeigt.

Einstellungen der Richtlinie "Endbenutzerüberwachung"

Der Abschnitt "Endbenutzerüberwachung" enthält Richtlinien zum Messen von Sitzungsnetzwerkverkehr.

ICA-Roundtripberechnung

Mit dieser Einstellung legen Sie fest, ob ICA-Roundtripberechnungen für aktive Verbindungen durchgeführt werden.

Standardmäßig sind die Berechnungen für aktive Verbindungen aktiviert.

Standardmäßig wird die Initiierung des ICA-Roundtripmessung verzögert, bis Netzwerkverkehr auf eine Benutzeraktion hinweist. Diese Verzögerung kann eine unbestimmte Länge haben und verhindert, dass die ICA-Roundtripmessung der einzige Grund für den ICA-Verkehr ist

Intervall für ICA-Roundtripberechnung

Mit dieser Einstellung geben Sie die Häufigkeit an, in Sekunden, mit der ICA-Roundtripberechnungen durchgeführt werden

Standardmäßig wird der ICA-Roundtrip alle 15 Sekunden berechnet.

ICA-Roundtrip für Verbindungen im Leerlauf berechnen

Mit dieser Einstellung legen Sie fest, ob ICA-Roundtripberechnungen für Verbindungen im Leerlauf durchgeführt werden.

Standardmäßig werden Berechnungen nicht für Verbindungen im Leerlauf durchgeführt.

Standardmäßig wird die Initiierung des ICA-Roundtripmessung verzögert, bis Netzwerkverkehr auf eine Benutzeraktion hinweist. Diese Verzögerung kann eine unbestimmte Länge haben und verhindert, dass die ICA-Roundtripmessung der einzige Grund für den ICA-Verkehr ist

Einstellungen der Richtlinie "Enhanced Desktop Experience"

Der Abschnitt "Enhanced Desktop Experience" enthält Richtlinieneinstellungen zum Konfigurieren von virtuellen Desktops, damit diese aussehen wie lokale Windows 7-Desktops.

Enhanced Desktop Experience

Durch diese Einstellung werden Sitzungen auf Serverbetriebssystemen so konfiguriert, dass sie wie lokale Windows 7-Desktops aussehen, damit Benutzer in den Genuss einer verbesserten Desktopdarstellung kommen.

Standardmäßig ist diese Einstellung zugelassen und Sitzungen auf Serverbetriebssystemen sehen aus wie Windows 7-Desktops.

Wenn ein Benutzerprofil mit dem Design "Windows – klassisch" bereits auf dem virtuellen Desktop vorhanden ist, wird durch Aktivieren dieser Richtlinie nicht die verbesserte Desktopdarstellung für diesen Benutzer bereitgestellt. Wenn sich ein Benutzer, dessen Benutzerprofil mit einem Windows 7-Design konfiguriert ist, bei einem virtuellen Desktop unter Windows Server 2012 anmeldet, für den diese Richtlinie deaktiviert oder nicht konfiguriert ist, wird eine Fehlermeldung angezeigt, die angibt, dass das Design nicht angewendet werden kann.

In beiden Fällen kann das Problem durch Zurücksetzen des Benutzerprofils gelöst werden.

Wenn die Richtlinie auf einem virtuellen Desktop mit aktiven Benutzersitzungen deaktiviert wird, sind Aussehen und Verhalten von Sitzungen nicht mit der Desktopdarstellung von Windows 7 und Windows –klassisch konsistent. Wenn Sie dies vermeiden möchten, starten Sie den virtuellen Desktop neu, nachdem Sie den Status der Richtlinie geändert haben. Sie müssen auch sämtliche Roamingprofile auf dem virtuellen Desktop löschen. Citrix empfiehlt außerdem, alle anderen Benutzerprofile auf dem virtuellen Desktop zu löschen, um Inkonsistenzen zwischen Profilen zu vermeiden.

Wenn Sie Roamingbenutzerprofile in der Umgebung verwenden, stellen Sie sicher, dass das Feature "Enhanced Desktop

Experience" für alle virtuellen Desktops, die sich ein Profil teilen, entweder aktiviert oder deaktiviert ist.

Citrix rät davon ab, Roamingprofile zwischen virtuellen Desktops, auf denen Serverbetriebssysteme und Clientbetriebssysteme ausgeführt werden, freizugeben. Die Profile für Client- und Serverbetriebssysteme sind unterschiedlich und das Freigeben von Roamingprofilen zwischen beiden Systemtypen kann zu Inkonsistenzen in den Profileigenschaften führen, wenn ein Benutzer zwischen den Systemen wechselt.

Einstellungen der Richtlinie "Dateiumleitung"

Der Abschnitt "Dateiumleitung" enthält Richtlinieneinstellungen für die Clientlaufwerkzuordnung und die Clientlaufwerkoptimierung.

Clientlaufwerke automatisch verbinden

Mit dieser Einstellung legen Sie fest, ob die automatische Verbindung von Clientlaufwerken bei der Benutzeranmeldung zugelassen ist.

In der Standardeinstellung ist die automatische Verbindung zugelassen.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, müssen Sie auch die Einstellungen für die Laufwerktypen aktivieren, die automatisch verbunden werden. Konfigurieren Sie beispielsweise Optische Clientlaufwerke, damit CD-Laufwerke auf dem Clientgerät automatisch verbunden werden.

Verwandte Richtlinieneinstellungen

- Clientlaufwerkumleitung
- Clientdiskettenlaufwerke
- Optische Clientlaufwerke
- Lokale Clientfestplattenlaufwerke
- Clientnetzlaufwerke
- Clientwechsellaufwerke

Clientlaufwerkumleitung

Mit dieser Einstellung aktivieren oder deaktivieren Sie die Dateiumleitung von und zu Laufwerken auf dem Benutzergerät.

In der Standardeinstellung ist die Dateiumleitung aktiviert.

Wenn aktiviert, können Benutzer ihre Dateien auf allen Clientlaufwerken speichern. Wenn deaktiviert, wird jegliche Dateiumleitung verhindert, unabhängig von den Dateiumleitungseinstellungen für einzelne Laufwerkstypen, z. B. Clientdiskettenlaufwerke und Clientnetzlaufwerke.

Verwandte Richtlinieneinstellungen

- Clientdiskettenlaufwerke
- Optische Clientlaufwerke
- Lokale Clientfestplattenlaufwerke
- Clientnetzlaufwerke
- Clientwechsellaufwerke

Lokale Clientfestplattenlaufwerke

Mit dieser Einstellung legen Sie fest, ob Benutzer auf die lokalen Festplattenlaufwerke des Benutzergeräts zugreifen oder Dateien darauf speichern können.

In der Standardeinstellung ist der Zugriff auf lokale Festplattenlaufwerke zugelassen.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, müssen Sie auch für die Einstellung Clientlaufwerkumleitung die Option Zugelassen wählen. Wenn diese Einstellungen deaktiviert sind, können lokale Festplattenlaufwerke nicht zugeordnet werden und Benutzer können auch nicht manuell auf diese Laufwerke zugreifen – unabhängig von der Einstellung für Lokale Festplattenlaufwerke.

Konfigurieren Sie außerdem die Einstellung Clientlaufwerke automatisch verbinden, damit lokale Festplattenlaufwerke automatisch verbunden werden.

Clientdiskettenlaufwerke

Mit dieser Einstellung legen Sie fest, ob Benutzer auf die Diskettenlaufwerke des Benutzergeräts zugreifen oder Dateien darauf speichern können.

In der Standardeinstellung ist der Zugriff auf Diskettenlaufwerke zugelassen.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, müssen Sie auch für die Einstellung Clientlaufwerkumleitung die Option Zugelassen wählen. Wenn diese Einstellungen deaktiviert sind, können Diskettenlaufwerke nicht zugeordnet werden und Benutzer können auch nicht manuell auf diese Laufwerke zugreifen – unabhängig von der Einstellung für Clientdiskettenlaufwerke.

Konfigurieren Sie außerdem die Einstellung Clientlaufwerke automatisch verbinden, damit Diskettenlaufwerke automatisch verbunden werden.

Clientnetzlaufwerke

Mit dieser Einstellung legen Sie fest, ob Benutzer auf die (remoten) Netzlaufwerke des Benutzergeräts zugreifen oder Dateien speichern können.

In der Standardeinstellung ist der Zugriff auf Netzlaufwerke zugelassen.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, müssen Sie auch für die Einstellung Clientlaufwerkumleitung die Option Zugelassen wählen. Wenn diese Einstellungen deaktiviert sind, können Netzlaufwerke nicht zugeordnet werden und Benutzer können auch nicht manuell auf diese Laufwerke zugreifen – unabhängig von der Einstellung für Clientnetzlaufwerke.

Konfigurieren Sie außerdem die Einstellung Clientlaufwerke automatisch verbinden, damit Netzlaufwerke automatisch verbunden werden.

Optische Clientlaufwerke

Mit dieser Einstellung legen Sie fest, ob Benutzer auf CD-, DVD- und BD-Laufwerke des Clientgeräts zugreifen oder Dateien dort speichern können.

In der Standardeinstellung ist der Zugriff auf optische Clientlaufwerke zugelassen.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, müssen Sie auch für die Einstellung Clientlaufwerkumleitung die

Option Zugelassen wählen. Wenn diese Einstellungen deaktiviert sind, können optische Clientlaufwerke nicht zugeordnet werden und Benutzer können auch nicht manuell auf diese Laufwerke zugreifen – unabhängig von der Einstellung für Optische Clientlaufwerke.

Konfigurieren Sie außerdem die Einstellung Clientlaufwerke automatisch verbinden, damit optische Clientlaufwerke automatisch verbunden werden.

Clientwechsellaufwerke

Mit dieser Einstellung legen Sie fest, ob Benutzer auf die USB-Laufwerke des Benutzergeräts zugreifen oder Dateien speichern können.

In der Standardeinstellung ist der Zugriff auf Clientwechsellaufwerke zugelassen.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, müssen Sie auch für die Einstellung Clientlaufwerkumleitung die Option Zugelassen wählen. Wenn diese Einstellungen deaktiviert sind, können Clientwechsellaufwerke nicht zugeordnet werden und Benutzer können auch nicht manuell auf diese Laufwerke zugreifen – unabhängig von der Einstellung für Clientwechsellaufwerke.

Konfigurieren Sie außerdem die Einstellung Clientlaufwerke automatisch verbinden, damit Clientwechsellaufwerke automatisch verbunden werden.

Host-zu-Client-Umleitung

Mit dieser Einstellung aktivieren oder deaktivieren Sie Dateitypzuordnungen für URLs und manche Medieninhalte, damit sie auf dem Clientgerät geöffnet werden. Wenn deaktiviert, werden Inhalte auf dem Server geöffnet.

In der Standardeinstellung ist die Dateitypzuordnung deaktiviert.

Diese Art von URLs werden lokal geöffnet, wenn Sie die Einstellung aktivieren:

- Hypertext Transfer Protocol (HTTP)
- Secure Hypertext Transfer Protocol (HTTPS)
- Real Player und QuickTime (RTSP)
- Real Player und QuickTime (RTSPU)
- Ältere Real Player-URLs (PNM)
- Microsoft Media Server (MMS)

Clientlaufwerksbuchstaben erhalten

Mit dieser Einstellung aktivieren oder deaktivieren Sie, ob die Clientlaufwerksbuchstaben erhalten bleiben.

In der Standardeinstellung bleiben die Clientlaufwerksbuchstaben nicht erhalten.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, müssen Sie auch für die Einstellung Clientlaufwerkumleitung die Option Zugelassen wählen.

Schreibgeschützter Zugriff auf Clientlaufwerke

Diese Einstellung erlaubt oder verhindert, dass Benutzer und Anwendungen Dateien oder Ordner auf zugeordneten Clientlaufwerken erstellen oder ändern.

Standardmäßig können Dateien und Ordner auf zugeordneten Clientlaufwerken geändert werden.

Wenn die Einstellung auf Aktiviert gesetzt wird, ist Lesezugriff auf die Dateien und Verzeichnisse möglich.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, müssen Sie auch für die Einstellung Clientlaufwerkumleitung die Option Zugelassen wählen.

Umleitung spezieller Ordner

Mit dieser Einstellung legen Sie fest, ob Benutzer von Citrix Receiver und dem Webinterface ihre lokalen speziellen Ordner in einer Sitzung sehen, z. B. "Dokumente" und "Desktop".

In der Standardeinstellung ist die Umleitung spezieller Ordner zugelassen.

Diese Einstellung verhindert, dass jegliche Objekte, die durch eine Richtlinie gefiltert werden, die Umleitung spezieller Ordner verwenden. Einstellungen an anderer Stelle werden nicht beachtet. Wenn Sie diese Einstellung zulassen, werden verwandte Einstellungen im Webinterface oder in Citrix Receiver ignoriert.

Sie legen fest, welche Benutzer die Umleitung spezieller Ordner erhalten, indem Sie Zugelassen wählen und diese Einstellung in eine Richtlinie aufnehmen, die nach den Benutzern gefiltert wird, denen diese Funktion zur Verfügung stehen soll. Diese Einstellung überschreibt alle anderen Einstellungen für die Umleitung spezieller Ordner.

Die Umleitung spezieller Ordner interagiert mit dem Clientgerät. Daher verhindern Einstellungen, die den Benutzerzugriff auf lokale Festplatten untersagen, auch die Umleitung spezieller Ordner.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, müssen Sie auch für die Einstellung Lokale Clientfestplattenlaufwerke die Option Zugelassen wählen.

Asynchrones Schreiben verwenden

Mit dieser Einstellung aktivieren oder deaktivieren Sie asynchrones Schreiben auf Laufwerke.

Standardmäßig ist das asynchrone Schreiben deaktiviert.

Für Verbindungen über WANs, die normalerweise durch relativ hohe Bandbreite und hohe Latenz charakterisiert sind, können Sie durch asynchrone Schreibvorgänge die Dateiübertragungen und Schreibvorgänge auf Clientlaufwerke beschleunigen. Sollte jedoch ein Verbindungsfehler oder Datenträgerfehler auftreten, können die Clientdateien, die geschrieben werden, in einem nicht definierten Zustand enden. Dem Benutzer werden dann in einem Pop-up-Fenster die betroffenen Dateien angezeigt. Der Benutzer kann das Problem beheben, z. B. durch Neustart einer unterbrochenen Dateiübertragung bei der Wiederverbindung oder nach Beheben eines Datenträgerfehlers.

Citrix empfiehlt, dass asynchrone Schreibvorgänge auf Datenträgern nur für Benutzer implementiert werden, die eine Remoteverbindung mit guter Geschwindigkeit für die Dateiübertragungen benötigen, und die verlorene Dateien oder Daten problemlos wiederherstellen können, sollten Fehler bei der Verbindung oder dem Datenträger auftreten.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, müssen Sie auch für die Einstellung Clientlaufwerkumleitung die Option Zugelassen wählen. Wenn diese Einstellung deaktiviert ist, finden keine asynchronen Schreibvorgänge statt.

Einstellungen der Richtlinie "Flash-Umleitung"

Der Abschnitt "Flash-Umleitung" enthält Richtlinieneinstellungen für die Behandlung von Flash-Inhalten in Benutzersitzungen.

Flash-Beschleunigung

Aktiviert oder deaktiviert die Wiedergabe von Flash-Inhalten auf Benutzergeräten statt auf dem Server. Standardmäßig ist die clientseitige Wiedergabe von Flash-Inhalten aktiviert.

Hinweis: Diese Einstellung wird für die Legacy-Flash-Umleitung mit dem Citrix Online Plug-In 12.1 verwendet.

Wenn aktiviert, reduziert diese Einstellung die Netzwerk- und Serverlast, indem Flash-Inhalte auf dem Clientgerät wiedergegeben werden. Zusätzlich erreichen Sie mit der Einstellung Flash-URL-Kompatibilitätsliste, dass Flash-Inhalte von bestimmten Websites auf dem Server wiedergegeben werden.

Auf dem Benutzergerät muss die Einstellung HDX MediaStream Flash-Umleitung auf dem Benutzergerät aktivieren auch aktiviert sein.

Ist diese Einstellung deaktiviert, werden Flash-Inhalte von allen Websites, unabhängig von deren URLs, auf dem Server wiedergegeben. Damit nur die Flash-Inhalte bestimmter Websites auf dem Benutzergerät wiedergegeben werden, konfigurieren Sie die Einstellung Flash-URL-Kompatibilitätsliste.

Flash-Hintergrundfarbenliste

Mit dieser Einstellung können Sie Schlüsselfarben für bestimmte URLs festlegen.

In der Standardeinstellung sind keine Schlüsselfarben angegeben.

Schlüsselfarben werden hinter clientseitig wiedergegebenem Flash angezeigt und unterstützen die visuelle Erkennung von Bereichen. Die angegebene Schlüsselfarbe sollte selten sein, sonst funktioniert die visuelle Bereichserkennung ggf. nicht richtig.

Gültige Eingaben bestehen aus einer URL (mit optionalen Platzhaltern am Anfang und Ende), der ein 24-Bit-RGB-Farbhexcode folgt. Zum Beispiel: `http://citrix.com 000003`

Flash-Abwärtskompatibilität

Mit dieser Einstellung aktivieren oder deaktivieren Sie die Verwendung von Originalfeatures für die Legacy-Flash-Umleitung mit älteren Versionen von Citrix Receiver (früher Citrix Online Plug-In).

Standardmäßig ist diese Einstellung aktiviert.

Auf dem Benutzergerät muss die Einstellung HDX MediaStream Flash-Umleitung auf dem Benutzergerät aktivieren auch aktiviert sein.

Flash-Umleitungsfeatures der zweiten Generation können mit Citrix Receiver 3.0 verwendet werden.

Legacyumleitungsfeatures werden mit dem Citrix Online Plug-In 12.1 unterstützt. Um sicherzustellen, dass die Flash-Umleitungsfeatures der zweiten Generation verwendet werden, muss die Flash-Umleitung der zweiten Generation auf dem Server und dem Benutzergerät aktiviert sein. Wenn die Legacyumleitung entweder auf dem Server oder dem Benutzergerät aktiviert ist, werden die Legacyumleitungsfeatures verwendet.

Flash-Standardverhalten

Mit dieser Einstellung wird das Standardverhalten für die Flash-Beschleunigung der zweiten Generation festgelegt.

Standardmäßig ist die Flash-Beschleunigung aktiviert.

Wählen Sie für die Konfiguration dieser Einstellung eine der folgenden Optionen:

- Flash-Beschleunigung aktivieren: Flash-Umleitung wird verwendet.
- Flash Player blockieren: Flash-Umleitung sowie serverseitiges Rendering werden nicht verwendet. Der Benutzer kann keine Flash-Inhalte anzeigen.
- Flash-Beschleunigung deaktivieren: Flash-Umleitung wird nicht verwendet. Der Benutzer kann auf dem Server wiedergegebene Flash-Inhalte anzeigen, wenn ein mit den Inhalten kompatibler Adobe Flash Player für Windows Internet Explorer auf dem Server installiert ist.

Diese Einstellung kann für einzelne Webseiten und Flash-Instanzen basierend auf der Einstellung Flash-URL-Kompatibilitätsliste außer Kraft gesetzt werden. Außerdem muss auf dem Benutzergerät die Einstellung HDX MediaStream Flash-Umleitung auf dem Benutzergerät aktivieren aktiviert sein.

Flash-Ereignisprotokollierung

Mit dieser Einstellung aktivieren oder deaktivieren Sie die Aufzeichnung von Flash-Ereignissen im Windows-Anwendungseignisprotokoll.

Standardmäßig ist die Protokollierung aktiviert.

Auf Computern unter Windows 7 oder Windows Vista wird ein spezielles Protokoll für die Flash-Umleitung im Knoten "Anwendungs- und Dienstprotokolle" angezeigt.

Flash - intelligentes Fallback

Mit dieser Einstellung aktivieren oder deaktivieren Sie automatische Versuche, das serverseitige Rendering für Flash Player-Instanzen zu verwenden, wenn das clientseitige Rendering entweder unnötig ist oder eine schlechte Benutzererfahrung ergibt.

Standardmäßig ist diese Einstellung aktiviert.

Flash-Latenzschwellenwert

Mit dieser Einstellung geben Sie einen Schwellenwert zwischen 0-30 Millisekunden an, um festzulegen, wo Adobe Flash-Inhalte wiedergegeben werden.

In der Standardeinstellung ist der Schwellenwert 30.

Beim Start wird die aktuelle Latenz zwischen Server und Benutzergerät von HDX MediaStream für Flash gemessen. Wenn die Latenz unter dem Schwellenwert liegt, werden Flash-Inhalte mit HDX MediaStream für Flash auf dem Benutzergerät wiedergegeben. Wenn die Latenz den Schwellenwert überschreitet, werden die Inhalte auf dem Netzwerkserver wiedergegeben, wenn Adobe Flash Player dort vorhanden ist.

Wenn Sie diese Einstellung aktivieren, muss die Einstellung Flash-Abwärtskompatibilität vorhanden und auf Zugelassen eingestellt sein.

Hinweis: Dies gilt nur, wenn HDX MediaStream Flash-Umleitung im Legacymodus verwendet wird.

URL-Liste für serverseitigen Flash-Inhaltsabruf

Mit dieser Einstellung geben Sie die Websites an, deren Flash-Inhalte auf den Server heruntergeladen und dann auf das Benutzergerät zur Wiedergabe übertragen werden.

In der Standardeinstellung sind keine Sites angegeben.

Diese Einstellung wird verwendet, wenn das Benutzergerät keinen direkten Internetzugang hat; der Server stellt die Verbindung bereit. Auf dem Benutzergerät muss die Einstellung Serverseitigen Inhaltsabruf aktivieren auch aktiviert sein.

Die Flash-Umleitung der zweiten Generation enthält ein Fallback für den serverseitigen Flash-Inhaltsabruf für SWF-Dateien. Wenn ein Benutzergerät Flash-Inhalte nicht von einer Website abrufen kann, und die Website in der URL-Liste für serverseitigen Flash-Inhaltsabruf angegeben ist, erfolgt automatisch der serverseitige Inhaltsabruf.

Hinzufügen von URLs zur Liste:

- Fügen Sie die URL der Flash-Anwendung und nicht die HTML-Seite auf oberster Ebene hinzu, die Flash Player initiiert.
- Verwenden Sie ein Sternchen (*) am Anfang oder am Ende der URL als Platzhalterzeichen.
- Verwenden Sie den Platzhalter am Ende, um alle untergeordneten URLs zuzulassen (<http://www.citrix.com/>).
- Die Präfixe <http://> und <https://> werden verwendet (falls vorhanden), sind jedoch für gültige Listeneinträge nicht vorgeschrieben.

Flash-URL-Kompatibilitätsliste

Mit dieser Einstellung geben Sie die Regeln an, die festlegen, ob die Wiedergabe von Flash-Inhalten auf bestimmten Websites auf dem Benutzergerät oder auf dem Server erfolgt oder ob sie blockiert wird.

In der Standardeinstellung sind keine Regeln angegeben.

Hinzufügen von URLs zur Liste:

- Sortieren Sie die Liste nach Priorität, sodass die wichtigsten URLs, Aktionen und Wiedergabeorte ganz oben stehen.
- Verwenden Sie ein Sternchen (*) am Anfang oder am Ende der URL als Platzhalterzeichen.
- Verwenden Sie den Platzhalter am Ende, um auf alle untergeordneten URLs zu verweisen (<http://www.citrix.com/>).
- Die Präfixe <http://> und <https://> werden verwendet (falls vorhanden), sind jedoch für gültige Listeneinträge nicht vorgeschrieben.
- Fügen Sie der Liste Websites hinzu, deren Flash-Inhalte nicht richtig auf dem Benutzergerät wiedergegeben werden, und wählen Sie entweder die Optionen Auf Server rendern oder Blockieren.

Einstellungen der Richtlinie "Grafiken"

Der Abschnitt "Grafiken" enthält Richtlinieneinstellungen, mit denen Sie steuern, wie Bilder in Benutzersitzungen behandelt werden.

Anzeigespeicherlimit

Mit dieser Einstellung geben Sie die maximale Größe des Videopuffers (in Kilobytes) für die Sitzung an.

Das Standardlimit ist 32768 Kilobytes.

Geben Sie einen Wert zwischen 128 und 131072 Kilobytes an. Verwenden einer größeren Farbtiefe und einer höheren Auflösung für Verbindungen erfordert mehr Speicher. Beim Erreichen des Speicherlimits wird die Anzeige gemäß der Einstellung Herabsetzungspräferenz für Anzeigemodus herabgesetzt.

Herabsetzungspräferenz für Anzeigemodus

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Diese Einstellung gibt an, ob die Farbtiefe oder die Auflösung zuerst herabgesetzt werden soll, wenn das Speicherlimit für die Sitzung erreicht wird.

Standardmäßig wird die Farbtiefe zuerst herabgesetzt.

Wenn das Speicherlimit der Sitzung erreicht wird, können Sie die Bildqualität verringern, indem Sie erst die Farbtiefe oder erst die Auflösung herabsetzen. Wird erst die Farbtiefe herabgesetzt, werden Bilder mit weniger Farben dargestellt. Wird erst die Auflösung herabgesetzt, werden Bilder mit weniger Pixel pro Zoll angezeigt.

Wenn Benutzer in dem Fall benachrichtigt werden sollen, dass entweder die Farbtiefe oder die Auflösung herabgesetzt werden muss, konfigurieren Sie die Einstellung Benutzer beim Herabsetzen des Anzeigemodus benachrichtigen.

Dynamische Fenstervorschau

Mit dieser Einstellung kann die Anzeige von Seamlessfenster in den Fenstervorschau Modi Flip, Flip 3D, Taskleistenvorschau und Einsehen aktiviert bzw. deaktiviert werden.

Standardmäßig ist diese Einstellung aktiviert.

Bildzwischenspeicherung

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung legen Sie fest, ob Bilder in Sitzungen zwischengespeichert werden. Falls erforderlich werden die Bilder in Abschnitten abgerufen, um das Ruckeln beim Bildlauf zu verringern.

Standardmäßig ist die Bildzwischenspeicherung aktiviert.

Legacygrafikmodus

Diese Einstellung deaktiviert die umfassende Grafikdarstellung und verwendet den Legacygrafikmodus, um die Skalierbarkeit über ein WAN oder eine mobile Verbindung zu verbessern.

Die Einstellung ist standardmäßig deaktiviert und die umfassende Grafikdarstellung wird verwendet.

Maximal zugelassene Farbtiefe

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung geben Sie die maximale Farbtiefe an, die für eine Sitzung zulässig ist.

Die Standardeinstellung für die maximal zulässige Farbtiefe ist 32 Bits pro Pixel.

Diese Einstellung gilt nur für ThinWire-Treiber und -Verbindungen. Sie gilt nicht für VDAs mit einem anderen Treiber als ThinWire für die primäre Anzeige, z. B. VDAs mit WDDM-Treiber (Windows Display Driver Model). Bei Desktopbetriebssystem-VDAs mit einem WDDM-Treiber als primären Anzeigetreiber hat diese Einstellung keine Auswirkung. Bei Windows-Serverbetriebssystem-VDAs mit WDDM-Treiber kann diese Einstellung verhindern, dass Benutzer eine Verbindung mit dem VDA herstellen.

Für eine hohe Farbtiefe ist mehr Speicher erforderlich. Damit die Farbtiefe herabgesetzt wird, wenn das Speicherlimit erreicht wurde, konfigurieren Sie die Einstellung Herabsetzungspräferenz für Anzeigemodus. Wird die Farbtiefe

herabgesetzt, werden Bilder mit weniger Farben dargestellt.

Benutzer beim Herabsetzen des Anzeigemodus benachrichtigen

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung erzielen Sie, dass Benutzer eine kurze Erklärung erhalten, wenn die Farbtiefe oder die Auflösung herabgesetzt wird.

Standardmäßig werden Benutzer nicht benachrichtigt.

Warteschlange und Verwerfen

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung werden Bilder in der Warteschlange verworfen, die durch ein anderes Bild ersetzt wurden.

Standardmäßig ist diese Einstellung aktiviert.

Dies verbessert die Reaktionszeit, wenn Grafiken an das Benutzergerät gesendet werden. Wenn Sie diese Einstellung konfigurieren, ruckeln Animationen möglicherweise, weil Frames ausgelassen werden.

Einstellungen der Richtlinie "Zwischenspeichern"

Der Abschnitt "Zwischenspeichern" enthält Einstellungen, mit denen Sie Bilddaten auf Benutzergeräten zwischenspeichern können, wenn Clientverbindungen eine beschränkte Bandbreite haben.

Schwellenwert für persistenten Cache

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung werden Bitmaps auf der Festplatte des Benutzergeräts zwischengespeichert. Dies ermöglicht eine Wiederverwendung von großen, oft verwendeten Bildern aus früheren Sitzungen.

Der Standardschwellenwert ist 3000000 Bits pro Sekunde.

Der Schwellenwert ist der Wert, unter dem das Feature "Permanentcache" angewendet wird. Beispielsweise werden mit dem Standardwert Bitmaps auf der Festplatte des Benutzergeräts zwischengespeichert, wenn die Bandbreite unter 3000000 Bit/s liegt.

Einstellungen der Richtlinie "Keep-Alive"

Der Abschnitt "Keep-Alive" enthält Richtlinieneinstellungen für die Verwaltung der ICA-Keep-Alive-Meldungen.

ICA-Keep-Alive - Timeout

Mit dieser Einstellung geben Sie die Anzahl der Sekunden zwischen aufeinanderfolgenden ICA-Keep-Alive-Meldungen an.

Das Standardintervall zwischen Keep-Alive-Meldungen ist 60 Sekunden.

Geben Sie ein Intervall zwischen 1-3600 Sekunden an, in dem ICA-Keep-Alive-Meldungen gesendet werden. Konfigurieren Sie diese Einstellung nicht, wenn Sie eine Netzwerküberwachungssoftware zum Schließen inaktiver Verbindungen verwenden.

ICA-Keep-Alives

Mit dieser Einstellung legen Sie fest, ob ICA-Keep-Alive-Meldungen in regelmäßigen Abständen gesendet werden sollen.

Standardmäßig werden keine Keep-Alive-Meldungen gesendet.

Wenn Sie diese Einstellung aktivieren, wird verhindert, dass unterbrochene Verbindungen getrennt werden. Wenn der Server keine Aktivität feststellt, verhindert diese Einstellung, dass die Sitzung durch die Remotedesktopdienste getrennt wird. Der Server sendet alle paar Sekunden Keep-Alive-Meldungen, um zu ermitteln, ob die Sitzung aktiv ist. Wenn die Sitzung nicht mehr aktiv ist, wird die Sitzung vom Server als "Getrennt" gekennzeichnet.

ICA-Keep-Alive funktioniert nicht, wenn Sie die Sitzungszuverlässigkeit verwenden. Konfigurieren Sie daher ICA-Keep-Alive nur für Verbindungen, die die Sitzungszuverlässigkeit nicht verwenden.

Verwandte Richtlinieneinstellungen

Sitzungszuverlässigkeit - Verbindungen

Einstellungen der Richtlinie "Lokaler App-Zugriff"

Der Abschnitt "Lokaler App-Zugriff" enthält Richtlinieneinstellungen, mit denen Sie die Integration lokal installierter Anwendungen mit gehosteten Anwendungen in einer gehosteten Desktopumgebung konfigurieren können.

Lokalen App-Zugriff zulassen

Mit dieser Einstellung legen Sie fest, ob die Integration lokal installierter Anwendungen mit gehosteten Anwendungen in einer gehosteten Desktopumgebung zugelassen oder verweigert werden soll.

Wenn ein Benutzer eine lokal installierte Anwendung startet, wirkt es so, als ob diese auf dem virtuellen Desktop des Benutzers ausgeführt würde, obwohl sie tatsächlich lokal ausgeführt wird.

Standardmäßig ist der lokale App-Zugriff nicht zulässig.

URL-Umleitungssperreliste

Mit dieser Einstellung geben Sie Websites an, die Ziel einer Weiterleitung sind und im lokalen Webbrowser gestartet werden sollen. Dies kann auch Websites umfassen, für die Gebietsschema-Informationen erforderlich sind (z. B. msn.com oder newsgoogle.com) oder Websites mit reichhaltigen Medieninhalten, die besser auf dem Benutzergerät wiedergegeben werden.

In der Standardeinstellung sind keine Sites angegeben.

URL-Umleitungspositivliste

Mit dieser Einstellung geben Sie die Websites an, die in der Umgebung, in der sie gestartet werden, wiedergegeben werden sollen.

In der Standardeinstellung sind keine Sites angegeben.

Einstellungen der Richtlinie "Mobilerfahrung"

Der Abschnitt "Mobilerfahrung" enthält Richtlinieneinstellungen für die Handhabung des Citrix Mobility Packs.

Automatische Anzeige der Tastatur

Diese Einstellung aktiviert oder deaktiviert die automatische Anzeige der Tastatur auf Bildschirmen von Mobilgeräten.

Standardmäßig ist die automatische Anzeige der Tastatur deaktiviert.

Für Fingereingabe optimierten Desktop starten

Diese Einstellung bestimmt das allgemeine Verhalten der Receiver-Benutzeroberfläche, indem sie eine für die Fingereingabe optimierte Benutzeroberfläche zulässt oder verweigert, die für Tablet-Geräte optimiert ist.

Standardmäßig wird eine für die Fingereingabe optimierte Benutzeroberfläche verwendet.

Stellen Sie diese Richtlinie auf Nicht zugelassen, um nur die Windows-Benutzeroberfläche zu verwenden.

Kombinationsfelder remoten

Diese Einstellung bestimmt die Typen von Kombinationsfeldern, die in Sitzungen auf mobilen Geräten angezeigt werden können. Stellen Sie diese Richtlinie auf Zugelassen ein, um das geräteeigene Kombinationsfeld-Steuererelement anzuzeigen. Wenn diese Einstellung zugelassen ist, kann ein Benutzer eine Sitzungseinstellung in Receiver für iOS ändern und das Windows-Kombinationsfeld verwenden.

Standardmäßig wird die Funktion zum Remoten von Kombinationsfeldern verweigert.

Einstellungen der Richtlinie "Multimedia"

Der Abschnitt "Multimedia" enthält Richtlinieneinstellungen, mit denen Sie das Streaming von Audio und Video in Benutzersitzungen verwalten.

Videoqualität beschränken

Mit dieser Einstellung geben Sie die maximale Videoqualitätsstufe für eine HDX-Verbindung an. Wird die Einstellung konfiguriert, dann wird die Videoqualität auf den angegebenen Wert beschränkt, sodass die Dienstqualität für Multimedia in der Umgebung gewährleistet ist.

Standardmäßig ist diese Einstellung nicht konfiguriert.

Zum Festlegen der maximalen Qualität wählen Sie eine der folgenden Optionen:

- 1080p/8,5 MBit/s
- 720p/4,0 MBit/s
- 480p/720 KBit/s
- 380p/400 KBit/s
- 240p/200 KBit/s

Hinweis: Die gleichzeitige Wiedergabe mehrerer Videos auf einem Server verbraucht viele Ressourcen und kann die Skalierbarkeit des Servers beeinträchtigen.

Multimediakonferenzen

Mit dieser Einstellung legen Sie fest, ob die Unterstützung von Videokonferenzanwendungen zugelassen ist.

Standardmäßig ist die Unterstützung für Videokonferenzen zugelassen.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, muss die Einstellung Windows Media-Umleitung vorhanden und auf Zugelassen gesetzt sein.

Für Multimediatelefonien müssen folgende Bedingungen erfüllt sein:

- Der Gerätetreiber des Herstellers für die in der Multimediatelefonie verwendete Webcam muss installiert sein.
- Die Webcam muss mit dem Clientgerät verbunden werden, bevor eine Videokonferenzsitzung initiiert wird. Der Server verwendet zu jedem Zeitpunkt nur eine installierte Webcam. Wenn mehrere Webcams auf dem Benutzergerät installiert sind, versucht der Server nacheinander jede Webcam zu verwenden, bis eine Videokonferenzsitzung steht.
- In der Umgebung muss ein Office Communicator-Server vorhanden sein.
- Die Office Communicator-Clientsoftware muss auf dem Server veröffentlicht sein.

Optimierung von Windows Media-Multimediaumleitung über WAN

Mit dieser Einstellung aktivieren Sie die Multimediatelefonierung in Echtzeit. Damit wird Audio- und Videostreaming für mobile Geräte ermöglicht und die Benutzererfahrung durch eine verbesserte Übermittlung von Windows Media-Inhalt über WAN optimiert.

Standardmäßig wird die Bereitstellung von Windows Media-Inhalt über das WAN optimiert.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, muss die Einstellung Windows Media-Umleitung vorhanden und auf Zugelassen gesetzt sein.

Wenn diese Einstellung aktiviert ist, wird die Multimediatelefonierung nach Bedarf automatisch bereitgestellt, sodass Audio- und Videostreaming zur Verbesserung der Benutzererfahrung auch bei schlechten Netzwerkbedingungen ermöglicht wird.

GPU für die Optimierung von Windows Media-Multimediaumleitung über WAN verwenden

Mit dieser Einstellung wird die Transcodierung von Multimediatelefonen in Echtzeit im Grafikprozessor (GPU) des Delivery Agent ermöglicht, um die Serverskalierbarkeit zu verbessern. Die GPU-Transcodierung ist nur verfügbar, wenn der Delivery Agent eine unterstützte GPU für die Hardwarebeschleunigung hat. Andernfalls erfolgt die Transcodierung automatisch in der CPU.

Hinweis: GPU-Transcodierung wird nur von NVIDIA-GPUs unterstützt.

Standardmäßig ist die Verwendung der GPU auf dem Delivery Agent zum Optimieren der Bereitstellung von Windows Media-Inhalt über das WAN nicht zulässig.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, stellen Sie sicher, dass die Einstellungen Windows Media-Umleitung und Optimierung von Windows Media-Multimediaumleitung über WAN vorhanden und auf Zugelassen gesetzt sind.

Clientseitiger Abruf von Windows Media-Inhalten

Diese Einstellung ermöglicht das Streamen von Multimediatelefonen anstatt über den Hostserver direkt vom Quellenanbieter im Internet oder Intranet auf Benutzergeräte.

Standardmäßig ist das direkte Streaming von Multimediatelefonen vom Quellenanbieter auf Benutzergeräte zulässig.

Zulassen dieser Einstellung verbessert die Netzwerknutzung und Serverskalierbarkeit durch Verschieben der gesamten Medienverarbeitung vom Hostserver auf das Benutzergerät. Dadurch wird auch das Erfordernis der Installation eines

erweiterten Multimedia-Frameworks, z. B. von Microsoft DirectShow oder Media Foundation, auf Benutzergeräten hinfällig. Diese müssen lediglich eine Datei von einer URL abspielen können.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, muss die Einstellung Windows Media-Umleitung vorhanden und auf Zugelassen gesetzt sein. Wenn diese Einstellung deaktiviert ist, ist das direkte Streaming von Multimediadateien auf Benutzergeräte ebenfalls deaktiviert.

Windows Media-Umleitung

Diese Einstellung steuert und optimiert die Art und Weise, mit der Streamingaudio und -video Benutzern von Servern bereitgestellt wird.

Standardmäßig ist das Streamen von Audio und Video für Benutzer zulässig.

Zulassen dieser Einstellung erhöht die Qualität von auf dem Server wiedergegebenem Audio und Video auf ein Niveau, das dem der lokalen Wiedergabe auf dem Benutzergerät entspricht. Der Server streamt Multimediainhalte komprimiert im Originalformat zum Client, Dekomprimierung und Wiedergabe der Medien übernimmt das Benutzergerät.

Die Windows Media-Umleitung optimiert Multimediadateien, die mit Codecs verschlüsselt sind, die den Standards von Microsoft DirectShow, DirectX Media Objects (DMO) und Media Foundation entsprechen. Um eine Multimediadatei wiederzugeben, muss ein mit dem Codierungsformat der Multimediadatei kompatibler Codec auf dem Benutzergerät vorhanden sein.

In der Standardeinstellung ist Audio auf Citrix Receiver deaktiviert. Wenn Benutzer Multimedia-Anwendungen in ICA-Sitzungen ausführen können, aktivieren Sie Audio oder geben Sie den Benutzern in der Receiver-Benutzeroberfläche die Berechtigung, Audio zu aktivieren.

Wählen Sie Verweigert nur, wenn die Wiedergabe von Medien mit der Windows Media-Umleitung schlechter zu sein scheint, als mit der ICA-Komprimierung und regulärem Audio. Dies ist selten, kann aber mit geringer Bandbreite vorkommen, beispielsweise bei Medien, in denen die Schlüsselbilder (Keyframes) sehr weit auseinander liegen.

Windows Media-Umleitungspuffergröße

Mit dieser Einstellung geben Sie für die Multimediabeschleunigung eine Puffergröße zwischen 1 und 10 Sekunden an.

Die Standardpuffergröße ist 5 Sekunden.

Verwendung von Windows Media-Umleitungspuffergröße

Mit dieser Einstellung aktivieren oder deaktivieren Sie die Verwendung der unter Windows Media-Umleitungspuffergröße angegebenen Puffergröße.

In der Standardeinstellung wird die angegebene Puffergröße nicht verwendet.

Wenn diese Einstellung deaktiviert oder die Einstellung für die Windows Media-Umleitungspuffergröße nicht konfiguriert ist, verwendet der Server den Standardwert für die Puffergröße (5 Sekunden).

Einstellungen der Richtlinie "Multistreamverbindungen"

Im Abschnitt "Multistreamverbindungen" finden Sie Richtlinieneinstellungen zum Verwalten der Quality-of-Service-Priorität (QoS) für mehrere ICA-Verbindungen in einer Sitzung.

Audio über UDP

Mit dieser Einstellung legen Sie fest, ob Audio über UDP auf dem Server zugelassen wird.

Standardmäßig ist Audio über UDP auf dem Server zugelassen.

Wenn diese Einstellung aktiviert ist, wird ein UDP-Port auf dem Server geöffnet, sodass alle Verbindungen, die zur Verwendung von Audio über UDP - Real-time Transport konfiguriert sind, unterstützt werden.

Audio-UDP-Portbereich

Mit dieser Einstellung geben Sie den Bereich der Portnummern an, die vom VDA für den Austausch von Audiopaketen mit dem Benutzergerät verwendet werden.

Die Standardeinstellung ist 16500,16509.

Der VDA versucht, jedes UDP-Portpaar für den Austausch von Daten mit dem Benutzergerät zu verwenden. Dabei wird mit dem Port, der die niedrigste Nummer hat, begonnen und die Zahl für jeden folgenden Versuch um 2 erhöht. Alle angegebenen Ports übernehmen eingehende und ausgehende Datenübertragungen.

Geben Sie einen Bereich im Format `niedrigste Portnummer, höchste Portnummer` an.

Multiportrichtlinie

Mit dieser Einstellung geben Sie die TCP-Ports an, die für den ICA-Verkehr verwendet werden sollen, und legen eine Netzwerkpriorität für jeden Port fest.

Standardmäßig hat der primäre Port (2598) eine hohe Priorität.

Wenn Sie Ports konfigurieren, können Sie die folgenden Prioritäten zuweisen:

- Sehr hoch
- Hoch
- Mittel
- Niedrig

Sie könnten eine sehr hohe Priorität zuweisen, wenn eine schnelle Reaktion in Echtzeit erforderlich ist, wie beispielsweise für Audio- und Videokonferenzen. Eine niedrige Priorität könnten Sie für Hintergrundprozesse zuweisen, wie beispielsweise das Drucken. Jeder Port muss eine eindeutige Priorität haben. Sie können also nicht eine sehr hohe Priorität sowohl für CGP-Port 1 als auch für CGP-Port 3 zuweisen.

Wenn Sie für einen Port keine Priorität einstellen möchten, setzen Sie den Wert für den Port auf 0. Sie können den primären Port nicht entfernen und Sie können seine Prioritätsstufe nicht ändern.

Wenn Sie diese Einstellung konfigurieren, starten Sie den Server neu. Diese Einstellung wird nur angewendet, wenn die Richtlinie Multistreamcomputereinstellung aktiviert ist.

Multistreamcomputereinstellung

Mit dieser Einstellung aktivieren oder deaktivieren Sie Multistream auf dem Server.

Standardmäßig ist Multistream deaktiviert.

Wenn Sie Citrix Cloudbridge mit Multistream-Unterstützung in Ihrer Umgebung verwenden, müssen Sie diese Einstellung nicht konfigurieren. Konfigurieren Sie diese Richtlinieneinstellung, wenn Sie Router von Drittanbietern oder Legacy-Branch Repeater verwenden, um die gewünschte Quality of Service (QoS) zu erzielen.

Wenn Sie diese Einstellung konfigurieren, starten Sie den Server neu, um sicherzustellen, dass die Änderungen wirksam werden.

Wichtig: Das Verwenden dieser Richtlinieneinstellung mit den Richtlinieneinstellungen für das Bandbreitenlimit, wie beispielsweise Bandbreitenlimit für Sitzung insgesamt kann zu unerwarteten Ergebnissen führen. Wenn Sie diese Einstellung in eine Richtlinie aufnehmen, stellen Sie sicher, dass Sie keine Bandbreitenlimit-Einstellungen einschließen.

Multistreambenutzereinstellung

Mit dieser Einstellung aktivieren oder deaktivieren Sie Multistream auf dem Benutzergerät.

Standardmäßig ist Multistream für alle Benutzer deaktiviert.

Diese Einstellung wird nur auf Hosts angewendet, für die die Richtlinie Multistreamcomputereinstellung aktiviert ist.

Wichtig: Das Verwenden dieser Richtlinieneinstellung mit den Richtlinieneinstellungen für das Bandbreitenlimit, wie beispielsweise Bandbreitenlimit für Sitzung insgesamt kann zu unerwarteten Ergebnissen führen. Wenn Sie diese Einstellung in eine Richtlinie aufnehmen, stellen Sie sicher, dass Sie keine Bandbreitenlimit-Einstellungen einschließen.

Einstellungen der Richtlinie "Portumleitung"

Der Abschnitt "Portumleitung" enthält Richtlinieneinstellungen für die LPT- und COM-Portzuordnung auf dem Client.

Hinweis: Konfigurieren Sie diese Einstellung für Virtual Delivery Agent 7.x über die Registrierung. Weitere Informationen finden Sie unter [Konfigurieren von COM-Port- und LPT-Portumleitungseinstellungen in der Registrierung](#).

Client-COM-Ports automatisch verbinden

Mit dieser Einstellung aktivieren oder deaktivieren Sie die automatische Verbindung von COM-Ports auf dem Benutzergerät, wenn Benutzer sich bei der Site anmelden.

Standardmäßig werden COM-Ports nicht automatisch verbunden.

Client-LPT-Ports automatisch verbinden

Mit dieser Einstellung aktivieren oder deaktivieren Sie die automatische Verbindung von LPT-Ports auf dem Benutzergerät, wenn Benutzer sich bei der Site anmelden.

Standardmäßig werden LPT-Ports des Clients automatisch verbunden.

Client-COM-Portumleitung

Mit dieser Einstellung legen Sie fest, ob der Zugriff auf COM-Ports des Benutzergeräts zulässig ist.

Standardmäßig ist die COM-Portumleitung nicht zugelassen.

Verwandte Richtlinieneinstellungen

- Bandbreitenlimit für COM-Portumleitung
- Bandbreitenlimit für COM-Portumleitung (Prozent)

Client-LPT-Portumleitung

Mit dieser Einstellung legen Sie fest, ob der Zugriff auf LPT-Ports des Benutzergeräts zulässig ist.

Standardmäßig ist die LPT-Portumleitung nicht zugelassen.

LPT-Ports werden nur von Legacyanwendungen verwendet, die Druckaufträge an LPT-Ports senden und nicht an die Druckerobjekte auf dem Benutzergerät. Die meisten Geräte können heute Druckaufträge and Druckerobjekte senden. Diese Richtlinieneinstellung ist nur für Server erforderlich, auf denen Legacyanwendungen gehostet werden, die für das Drucken LPT-Ports verwenden.

Verwandte Richtlinieneinstellungen

- Bandbreitenlimit für LPT-Portumleitung
- Bandbreitenlimit für LPT-Portumleitung (Prozent)

Einstellungen der Richtlinie "Drucken"

Der Abschnitt "Drucken" enthält Richtlinieneinstellungen für die Verwaltung des Clientdrucks.

Clientdruckerumleitung

Mit dieser Einstellung legen Sie fest, ob Clientdrucker einem Server zugeordnet werden können, wenn sich ein Benutzer an einer Sitzung anmeldet.

Standardmäßig ist die Clientdruckerzuordnung zugelassen.

Verwandte Richtlinieneinstellungen

Clientdrucker automatisch erstellen

Standarddrucker

Mit dieser Einstellung geben Sie an, wie der Standarddrucker in einer ICA-Sitzung ermittelt wird.

Standardmäßig wird der aktuelle Standarddrucker auf dem Clientgerät als Standarddrucker in der Sitzung verwendet.

Mit Standarddrucker des Benutzers nicht anpassen werden die aktuellen Einstellungen für den Standarddrucker in den Remotedesktopdiensten oder im Windows-Benutzerprofil verwendet. Wenn Sie diese Option auswählen, wird der Standarddrucker nicht im Profil gespeichert und ändert sich nicht entsprechend den anderen Sitzungs- oder Clienteigenschaften. Der Standarddrucker in einer Sitzung ist der erste Drucker, der in der Sitzung automatisch erstellt wird. Das ist:

- Der erste Drucker, der lokal auf dem Windows-Server unter Systemsteuerung > Geräte und Drucker hinzugefügt wurde.
- Der erste automatisch erstellte Drucker, wenn auf dem Server keine Drucker lokal hinzugefügt wurden.

Verwenden Sie diese Option, um Benutzern über Profileinstellungen den nächstgelegenen Drucker anzubieten (Proximitydrucken).

Druckerzuweisungen

Diese Einstellung bietet eine Alternative zu den Einstellungen Standarddrucker und Sitzungsdrucker. Mit den einzelnen Einstellungen für Standarddrucker und Sitzungsdrucker können Sie das Verhalten einer Site, einer großen Gruppe oder einer Organisationseinheit konfigurieren. Mit der Einstellung Druckerzuweisungen weisen Sie eine große Gruppe Drucker mehreren Benutzern zu.

Mit dieser Einstellung geben Sie an, wie der Standarddrucker auf den aufgeführten Benutzergeräten in einer Sitzung ermittelt wird.

Standardmäßig wird der aktuelle Standarddrucker auf dem Clientgerät als Standarddrucker in der Sitzung verwendet.

Mit dieser Einstellung geben Sie außerdem die Netzwerkdrucker an, die in einer Sitzung für jedes Benutzergerät automatisch erstellt werden sollen. In der Standardeinstellung sind keine Drucker angegeben.

- Beim Einstellen des Standarddruckerwerts:

Wenn Sie den aktuellen Standarddrucker für das Benutzergerät verwenden möchten, wählen Sie Nicht anpassen.

Mit Do not adjust werden die aktuellen Einstellungen für den Standarddrucker in den Remotedesktopdiensten oder im Windows-Benutzerprofil verwendet. Wenn Sie diese Option auswählen, wird der Standarddrucker nicht im Profil gespeichert und ändert sich nicht entsprechend den anderen Sitzungs- oder Clienteigenschaften. Der Standarddrucker in einer Sitzung ist der erste Drucker, der in der Sitzung automatisch erstellt wird. Das ist:

- Der erste Drucker, der lokal auf dem Windows-Server unter Systemsteuerung > Geräte und Drucker hinzugefügt wurde.
- Der erste automatisch erstellte Drucker, wenn auf dem Server keine Drucker lokal hinzugefügt wurden.
- Beim Einstellen des Sitzungsdruckerwerts:
 - Um Drucker hinzuzufügen, geben Sie den UNC-Pfad des Druckers ein, der automatisch erstellt werden soll. Nach dem Hinzufügen des Druckers können Sie angepasste Einstellungen für die aktuelle Sitzung bei jeder Anmeldung anwenden.

Ereignisprotokollpräferenz für die automatische Druckererstellung

Mit dieser Einstellung geben Sie an, welche Ereignisse bei der automatischen Druckererstellung protokolliert werden. Sie haben die Option, keine Fehler oder Warnungen, nur Fehler oder Fehler und Warnungen zu protokollieren.

Standardmäßig werden Fehler und Warnungen protokolliert.

Ein Beispiel für eine Warnung ist ein Ereignis, bei dem der native Druckertreiber für einen Drucker nicht installiert werden konnte und stattdessen der universelle Druckertreiber installiert wurde. Damit der universelle Druckertreiber in diesem Szenario verwendet werden kann, stellen Sie für Verwendung universeller Druckertreiber entweder Nur universelles Drucken verwenden oder Universelles Drucken nur verwenden, wenn angeforderter Treiber nicht verfügbar ist ein.

Sitzungsdrucker

Mit dieser Einstellung geben Sie die Netzwerkdrucker an, die in einer ICA-Sitzung automatisch erstellt werden sollen.

In der Standardeinstellung sind keine Drucker angegeben.

Um Drucker hinzuzufügen, geben Sie den UNC-Pfad des Druckers ein, der automatisch erstellt werden soll. Nach dem Hinzufügen des Druckers können Sie angepasste Einstellungen für die aktuelle Sitzung bei jeder Anmeldung anwenden.

Warten bis Drucker erstellt sind (Desktop)

Mit dieser Einstellung legen Sie fest, ob es eine Verzögerung bei der Sitzungsverbindung geben soll, sodass Desktopdrucker automatisch erstellt werden können.

Standardmäßig findet keine Verbindungsverzögerung statt.

Einstellungen der Richtlinie "Clientdrucker"

Der Abschnitt "Clientdrucker" enthält Richtlinieneinstellungen für Clientdrucker, einschließlich solcher zur automatischen Erstellung von Clientdruckern, zum Speichern von Druckereigenschaften und zum Verbinden mit Druckservern.

Clientdrucker automatisch erstellen

Mit dieser Einstellung geben Sie die Clientdrucker an, die automatisch erstellt werden. Diese Einstellung überschreibt die Standardeinstellungen für die automatische Clientdruckererstellung.

Standardmäßig werden alle Clientdrucker automatisch erstellt.

Diese Einstellung gilt nur, wenn die Einstellung Clientdruckerumleitung vorhanden und zugelassen ist.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, wählen Sie eine der Optionen:

- Mit Alle Clientdrucker automatisch erstellen werden alle Drucker auf dem Clientgerät erstellt.
- Mit Nur Standarddrucker des Clients automatisch erstellen wird der Drucker automatisch erstellt, der auf dem Clientgerät als Standarddrucker angegeben wurde.
- Mit Nur lokale Clientdrucker (keine Netzwerkdrucker) automatisch erstellen werden nur die Drucker automatisch erstellt, die über einen LPT-, COM-, USB-, TCP/IP- oder anderen lokalen Port direkt mit dem Clientgerät verbunden sind.
- Mit Clientdrucker nicht automatisch erstellen wird die automatische Erstellung von Clientdruckern beim Anmelden der Benutzer deaktiviert. Die Remotedesktopdienste-Einstellungen für die automatische Erstellung von Clientdruckern überschreiben dann diese Einstellung in Richtlinien mit niedrigerer Priorität.

Generischen universellen Drucker automatisch erstellen

Mit dieser Einstellung aktivieren oder deaktivieren Sie die automatische Erstellung des generischen universellen Citrix Druckerobjekts für Sitzungen, in denen ein Benutzergerät verwendet wird, das mit der universellen Drucklösung kompatibel ist.

Standardmäßig wird das generische universelle Druckerobjekt nicht automatisch erstellt.

Verwandte Richtlinieneinstellungen

- Verwendung universeller Druckertreiber
- Priorität universeller Treiber

Clientdruckernamen

Mit dieser Einstellung legen Sie die Namenskonvention für automatisch erstellte Drucker fest.

Standardmäßig werden die Standardnamen der Drucker verwendet.

Wählen Sie Standarddruckernamen, um Druckernamen zu verwenden, die mit denen, die durch native Remotedesktopdienste erstellt werden, vergleichbar sind (z. B. "HPLaserJet 4 von Clientname in Sitzung 3").

Wählen Sie Legacydruckernamen, um Clientdruckernamen im alten Stil zuzulassen und die Rückwärtskompatibilität für Benutzer oder Gruppen zu erhalten, die MetaFrame Presentation Server 3.0 oder früher verwenden. Ein Beispiel für einen Legacydruckernamen ist: Client/Clientname#/HPLaserJet 4. Da diese Option geringere Sicherheit bietet, sollten Sie sie nur für Benutzer und Gruppen verwenden, für die die Rückwärtskompatibilität mit MetaFrame Presentation Server 3.0 oder früher erforderlich ist.

Hinweis: Diese Option wird nur für Abwärtskompatibilität mit älteren Versionen von XenApp und XenDesktop bereitgestellt.

Direkte Verbindungen zu Druckservern

Mit dieser Einstellung aktivieren oder deaktivieren Sie direkte Verbindungen vom virtuellen Desktop oder Server, auf dem Anwendungen gehostet werden, zu einem Druckserver für Clientdrucker in einer zugänglichen Netzwerkfreigabe.

Standardmäßig sind direkte Verbindungen aktiviert.

Aktivieren Sie direkte Verbindungen, wenn der Netzwerkdruckserver für virtuelle Desktops bzw. für Server, auf denen Anwendungen gehostet werden, nicht über ein WAN zugänglich ist. Direkte Verbindungen gestatten schnelleres Drucken, wenn der Netzwerkdruckserver und der virtuelle Desktop oder Anwendungsserver sich im gleichen LAN befinden.

Deaktivieren Sie direkte Verbindungen, wenn das Netzwerk über ein WAN verläuft oder hohe Latenz oder beschränkte Bandbreite aufweist. Druckaufträge werden durch das Benutzergerät und den Netzwerkdruckserver geleitet. Daten werden komprimiert an das Benutzergerät gesendet, es wird somit weniger Bandbreite bei der Übertragung der Daten über das WAN gebraucht.

Wenn zwei Netzwerkdrucker den gleichen Namen haben, wird der Drucker benutzt, der im gleichen Netzwerk ist wie der Client.

Druckertreiberzuordnung und -kompatibilität

Mit dieser Einstellung legen Sie Regeln für die Treiberersetzung bei automatisch erstellten Druckern fest.

In der Standardeinstellung sind keine Regeln angegeben.

Wenn Sie Regeln für die Treiberersetzung definieren, können Sie zulassen oder verhindern, dass Drucker mit dem angegebenen Treiber erstellt werden. Außerdem können Sie für erstellte Drucker nur universelle Druckertreiber zulassen. Bei der Treiberersetzung werden die Namen der Druckertreiber, die das Benutzergerät bereitstellt, überschrieben oder zugeordnet und ein äquivalenter Treiber auf dem Server wird ersetzt. So können Serveranwendungen auf Clientdrucker zuzugreifen, die denselben Treiber wie der Server, aber unterschiedliche Treibernamen verwenden.

Sie können eine Treiberzuordnung hinzufügen, eine bestehende Zuordnung bearbeiten, benutzerdefinierte Einstellungen für eine Zuordnung überschreiben oder die Reihenfolge der Treibereinträge in der Liste ändern. Um eine Zuordnung hinzuzufügen, geben Sie den Clientdruckertreibernamen an und wählen dann den Ersatztreiber auf dem Server aus.

Speicherung von Druckereigenschaften

Mit dieser Einstellung geben Sie an, ob die Druckereigenschaften gespeichert werden und wo.

Standardmäßig ermittelt das System, ob Druckereigenschaften auf dem Clientgerät (falls verfügbar) gespeichert werden oder im Benutzerprofil.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, wählen Sie eine der Optionen:

- Wählen Sie Nur auf dem Clientgerät speichern, wenn Sie ein vorgeschriebenes oder servergespeichertes Profil verwenden, das nicht gespeichert wird. Wählen Sie diese Option nur aus, wenn auf allen Servern in der Farm XenApp 5 oder höher ausgeführt wird und die Benutzer die Citrix Online Plug-In Versionen 9.x, 10.x, 11.x und 12.x oder Citrix Receiver 3.x verwenden.
- Wählen Sie Nur im Benutzerprofil speichern, wenn das System durch die Bandbreite (diese Option reduziert den Datenverkehr im Netzwerk) und die Anmeldegeschwindigkeit begrenzt ist oder die Benutzer Legacy-Plug-Ins verwenden. Bei dieser Option werden die Druckereigenschaften im Benutzerprofil auf dem Server gespeichert. Die Eigenschaften

werden nicht mit dem Clientgerät ausgetauscht. Verwenden Sie diese Option für MetaFrame Presentation Server 3.0 oder früher und MetaFrame Presentation Server Client 8.x oder früher. Dies gilt nur, wenn ein servergespeichertes Profil für Remotedesktopdienste (RDS) verwendet wird.

- Mit Nur im Profil speichern, wenn sie nicht auf dem Client gespeichert sind kann das System festlegen, wo die Druckereigenschaften gespeichert werden. Die Druckereigenschaften werden auf dem Clientgerät gespeichert, sofern es verfügbar ist, ansonsten im Benutzerprofil. Diese Option bietet zwar die größte Flexibilität, kann jedoch die Anmeldezeit verlangsamen und zusätzliche Bandbreite für die Systemprüfung verbrauchen.
- Druckereigenschaften nicht speichern verhindert das Speichern von Druckereigenschaften.

Gespeicherte und wiederhergestellte Clientdrucker

Mit dieser Einstellung aktivieren oder deaktivieren Sie das Speichern und Neuerstellen von Clientdruckern. Standardmäßig werden Clientdrucker automatisch gespeichert und automatisch wiederhergestellt.

Gespeicherte Drucker sind vom Benutzer erstellte Drucker, die beim Start der nächsten Sitzung wiederhergestellt werden. Wenn XenApp einen gespeicherten Drucker wiederhergestellt, werden alle Richtlinienereinstellungen außer Clientdrucker automatisch erstellen berücksichtigt.

Gespeicherte Drucker sind Drucker die von einem Administrator vollständig angepasst wurden und deren gespeicherter Zustand permanent mit einem Clientport verbunden ist.

Einstellungen der Richtlinie "Treiber"

Der Abschnitt "Treiber" enthält Richtlinienereinstellungen für Druckertreiber.

Automatische Installation von mitgelieferten Druckertreibern

Mit dieser Einstellung aktivieren oder deaktivieren Sie die automatische Installation von Druckertreibern vom standardmäßigen Windows-Treibersatz oder von Treiberpaketen, die auf dem Host mit folgendem Befehl bereitgestellt wurden: `pnputil.exe /a`.

Standardmäßig werden diese Treiber bei Bedarf installiert.

Priorität universeller Treiber

Mit dieser Einstellung geben Sie an, in welcher Reihenfolge die universellen Druckertreiber verwendet werden, angefangen mit dem ersten Eintrag in der Liste.

Standardmäßig ist die Prioritätsreihenfolge wie folgt:

- EMF
- XPS
- PCL5c
- PCL4
- PS

Sie können Treiber hinzufügen, bearbeiten oder entfernen und die Reihenfolge der Treiber in der Liste ändern.

Verwendung universeller Druckertreiber

Mit dieser Einstellung geben Sie an, wann universelles Drucken verwendet wird.

Standardmäßig wird universelles Drucken nur verwendet, wenn der angeforderte Treiber nicht verfügbar ist.

Universelles Drucken verwendet allgemeine Druckertreiber statt modellspezifischer Standardtreiber; dies verringert potentiell den Aufwand für die Treiberverwaltung auf Hostcomputern. Die Verfügbarkeit universeller Druckertreiber hängt von den Funktionen des Benutzergeräts, des Hosts und der Druckersoftware ab. In bestimmten Konfigurationen steht universelles Drucken möglicherweise nicht zur Verfügung.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, wählen Sie eine der Optionen:

- Mit Nur druckerspezifische Treiber verwenden verwendet der Clientdrucker nur die modellspezifischen Standardtreiber, die bei der Anmeldung automatisch erstellt wurden. Wenn der erforderliche Treiber nicht verfügbar ist, kann der Clientdrucker nicht automatisch erstellt werden.
- Nur universelles Drucken verwenden gibt an, dass keine modellspezifischen Standardtreiber verwendet werden. Nur universelle Druckertreiber werden zum Erstellen von Druckern verwendet.
- Mit Universelles Drucken nur verwenden, wenn angeforderter Treiber nicht verfügbar ist werden modellspezifische Standardtreiber für die Druckererstellung verwendet, wenn sie verfügbar sind. Wenn der Treiber auf dem Server nicht zur Verfügung steht, wird der Clientdrucker automatisch mit dem entsprechenden universellen Treiber erstellt.
- Mit Druckerspezifische Treiber nur verwenden, wenn universelles Drucken nicht verfügbar ist werden universelle Druckertreiber verwendet, wenn sie verfügbar sind. Wenn der Treiber auf dem Server nicht zur Verfügung steht, wird der Clientdrucker automatisch mit dem entsprechenden modellspezifischen Druckertreiber erstellt.

Einstellungen der Richtlinie "Universelles Drucken"

Der Abschnitt "Universelles Drucken" enthält Richtlinieneinstellungen für die Verwaltung des universellen Drucks.

Universelles Drucken - EMF-Verarbeitungsmodus

Mit dieser Einstellung steuern Sie die Verarbeitungsmethode für die EMF-Spooldatei auf dem Windows-Benutzergerät.

Standardmäßig werden EMF-Datensätze direkt zum Drucker gespoolt.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, wählen Sie eine der Optionen:

- EMF-Datensätze für Drucker neu verarbeiten erzwingt die Neuverarbeitung der EMF-Spooldatei und sendet sie durch das GDI-Teilsystem auf dem Benutzergerät. Sie können diese Einstellung für Treiber verwenden, für die eine EMF-Neuverarbeitung erforderlich ist, die jedoch nicht unbedingt in der Sitzung automatisch ausgewählt werden.
- Wenn Direkt zum Drucker spoolen mit dem universellen Citrix Druckertreiber verwendet wird, werden die EMF-Datensätze garantiert gespoolt und an das Benutzergerät für die Verarbeitung übergeben. Diese EMF-Spooldateien werden normalerweise direkt in die Spoolwarteschlange des Clients gesetzt. Für Drucker und Treiber, die mit dem EMF-Format kompatibel sind, ist dies die schnellste Druckmethode.

Universelles Drucken - Bildkomprimierungslimit

Mit dieser Einstellung geben Sie die maximale Qualität und minimale Komprimierung für Bilder an, die mit dem universellen Citrix Druckertreiber gedruckt werden.

Das Limit für Bildkomprimierung ist standardmäßig auf Beste Qualität (verlustfreie Komprimierung) gesetzt.

Wenn Keine Komprimierung ausgewählt ist, wird die Komprimierung nur für den EMF-Druck deaktiviert.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, wählen Sie eine der Optionen:

- Keine Komprimierung

- Beste Qualität (verlustfreie Komprimierung)
- Hohe Qualität
- Standardqualität
- Geringere Qualität (maximale Komprimierung)

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, die auch die Einstellung "Universelles Drucken - Optimierungsstandards" enthält, achten Sie auf Folgendes:

- Wenn die Komprimierungsstufe in der Einstellung Universelles Drucken - Komprimierungslimit niedriger ist als die in der Einstellung Universelles Drucken - Optimierungsstandards werden Bilder basierend auf der Einstellung Universelles Drucken - Komprimierungslimits komprimiert.
- Wenn die Komprimierung deaktiviert ist, haben die Optionen Gewünschte Bildqualität und Heavyweight-Komprimierung aktivieren in der Einstellung Universelles Drucken - Optimierungsstandards keine Auswirkung in der Richtlinie.

Universelles Drucken - Optimierungsstandards

Mit dieser Einstellung geben Sie die Standardwerte für die Druckoptimierung an, wenn der universelle Druckertreiber für eine Sitzung erstellt wurde.

- Mit Gewünschte Bildqualität geben Sie das standardmäßige Bildkomprimierungslimit an, das auf universelles Drucken angewendet wird. In der Standardeinstellung ist Standardqualität aktiviert, d. h. Benutzer können Bilder nur mit der Standardqualitäts- oder geringeren Qualitätskomprimierung drucken.
- Mit Heavyweight-Komprimierung aktivieren oder deaktivieren Sie das Verringern der Bandbreite unter den Komprimierungsgrad, der von Gewünschte Bildqualität festgelegt ist; Bildqualität geht nicht verloren. Standardmäßig ist die Heavyweight-Komprimierung deaktiviert.
- Mit den Einstellungen Zwischenspeichern von Bildern und Schriftarten legen Sie fest, ob Bilder und Schriftarten, die mehrmals im Druckdatenstrom vorhanden sind, zwischengespeichert werden. Sie stellen damit sicher, dass jedes eindeutige Bild oder jede Schriftart nur einmal zum Drucker gesendet wird. Standardmäßig werden eingebettete Bilder und Schriftarten zwischengespeichert. Hinweis: Diese Einstellungen gelten nur, wenn das Benutzergerät dieses Verhalten unterstützt.
- Mit Nicht-Administratoren können diese Einstellungen ändern legen Sie fest, ob Benutzer die Standardeinstellungen für die Druckoptimierung in einer Sitzung ändern können. Standardmäßig können Benutzer die Standardeinstellungen für die Druckoptimierung nicht ändern.

Hinweis: Alle diese Optionen werden für den EMF-Druck unterstützt. Für XPS-Druck wird nur die Option Gewünschte Bildqualität unterstützt.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, die auch die Einstellung "Universelles Drucken - Optimierungsstandards" enthält, achten Sie auf Folgendes:

- Wenn die Komprimierungsstufe in der Einstellung Universelles Drucken - Komprimierungslimit niedriger ist als die in der Einstellung Universelles Drucken - Optimierungsstandards werden Bilder basierend auf der Einstellung Universelles Drucken - Komprimierungslimits komprimiert.
- Wenn die Komprimierung deaktiviert ist, haben die Optionen Gewünschte Bildqualität und Heavyweight-Komprimierung aktivieren in der Einstellung Universelles Drucken - Optimierungsstandards keine Auswirkung in der Richtlinie.

Universelles Drucken - Vorschau-einstellung

Mit dieser Einstellung geben Sie an, ob die Druckvorschau für automatisch erstellte oder universelle Drucker verwendet werden soll.

Standardmäßig wird die Druckvorschau für automatisch erstellte oder generische universelle Drucker nicht verwendet.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, wählen Sie eine der Optionen:

- Druckvorschau für automatisch erstellte oder generische universelle Drucker nicht verwenden
- Druckervorschau nur für automatisch erstellte Drucker verwenden
- Druckvorschau nur für generische universelle Drucker verwenden
- Druckvorschau für automatisch erstellte und generische universelle Drucker verwenden

Universelles Drucken - Druckqualitätslimit

Diese Einstellung legt den Höchstwert für Punkte pro Zoll (dpi) zum Erstellen von Ausdrucken in einer Sitzung fest.

Standardmäßig ist Kein Limit aktiviert, d. h. Benutzer können die höchste Druckqualität auswählen, die vom Drucker zugelassen wird, mit dem sie eine Verbindung herstellen.

Wenn diese Einstellung konfiguriert ist, wird die maximale Druckqualität, die Benutzern zur Verfügung steht, hinsichtlich Ausgabeauflösung beschränkt. Sowohl die Druckqualität und die Druckqualitätsmerkmale des Druckers, mit dem sich die Benutzer verbinden, werden auf die konfigurierte Einstellung beschränkt. Beispiel: Wenn Mittlere Auflösung (600 dpi) konfiguriert ist, können Benutzer eine Ausgabe nur mit einer maximalen Qualität von 600 drucken, und die Einstellung "Druckqualität" auf der Registerkarte "Erweitert" im Dialogfeld "Universeller Drucker" enthält nur Auflösungseinstellungen bis zu "Mittlere Qualität (600 dpi).

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, wählen Sie eine der Optionen:

- Entwurf (150 dpi)
- Niedrige Auflösung (300 dpi)
- Mittlere Auflösung (600 dpi)
- Hohe Auflösung (1200 dpi)
- Kein Limit

Einstellungen der Richtlinie "Universeller Druckserver"

Der Abschnitt "Universeller Druckserver" enthält Richtlinieneinstellungen für die Behandlung des universellen Druckservers.

Universellen Druckserver aktivieren

Diese Einstellung aktiviert oder deaktiviert das Feature "Universeller Druckserver" auf dem virtuellen Desktop oder dem Server, auf dem Anwendungen gehostet werden. Wenden Sie die Richtlinie auf Organisationseinheiten an, die den virtuellen Desktop oder Server enthalten, auf dem Anwendungen gehostet werden.

Standardmäßig ist das Feature deaktiviert.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, wählen Sie eine der folgenden Optionen:

- **Aktiviert mit Fallback auf systemeigenen Windows-Remotedruck:** Netzwerkdruckerverbindungen werden nach Möglichkeit vom universellen Druckserver bedient. Wenn dieser nicht verfügbar ist, wird der Windows-Druckanbieter verwendet. Der Windows-Druckanbieter handhabt weiterhin alle Drucker, die vorher mit dem Windows-Druckanbieter erstellt wurden.
- **Aktiviert ohne Fallback auf systemeigenen Windows-Remotedruck:** Netzwerkdruckerverbindungen werden ausschließlich vom universellen Druckserver bedient. Wenn dieser nicht verfügbar ist, schlägt die Netzwerkdrucker Verbindung fehl. Mit dieser Einstellung wird der Netzwerkdruck über den Windows-Druckanbieter effektiv deaktiviert. Drucker, die vorher mit dem Windows-Druckanbieter erstellt wurden, werden nicht erstellt, solange eine Richtlinie mit dieser Einstellung aktiv ist.
- **Deaktiviert:** Das Feature "Universeller Druckserver" ist deaktiviert. Beim Herstellen einer Verbindung mit einem

Netzwerkdrucker, der einen UNC-Namen hat, wird keine Verbindung mit dem universellen Druckserver versucht.
Verbindungen mit Remotedruckern verwenden weiterhin den Windows-Remotedruck.

Port für Druckdatenstrom des universellen Druckservers (CGP)

Diese Einstellung gibt die Nummer des TCP-Ports an, die vom Druckdatenstrom-Listener (CGP) des universellen Druckservers verwendet wird. Wenden Sie diese Richtlinie nur für Organisationseinheiten an, die den Druckserver enthalten.

Die Standardeinstellung der Portnummer ist "7229".

Gültige Portnummern müssen im Bereich von 1 bis 65535 liegen.

Universeller Druckserver - Eingabebandbreitenlimit für Druckdatenstrom (KBit/s)

Diese Einstellung gibt das obere Limit (in Kilobit pro Sekunde) für die Übertragungsrate der Druckdaten an, die von jedem Druckauftrag mit CGP an den universellen Druckserver übergeben werden. Wenden Sie die Richtlinie auf Organisationseinheiten an, die den virtuellen Desktop oder Server enthalten, auf dem Anwendungen gehostet werden.

In der Standardeinstellung ist der Wert 0, was angibt, dass es kein oberes Limit gibt.

Port für universellen Druckserverwebdienst (HTTP/SOAP)

Diese Einstellung gibt die Nummer des TCP-Ports an, die vom Webdienstlistener des universellen Druckservers für eingehende HTTP/SOAP-Anforderungen verwendet wird. Stellen Sie sicher, dass Wert dieser Einstellung für die Organisationseinheit, die den Druckserver enthält und für die Organisationseinheit, die den virtuellen Desktop oder Server enthält, auf dem Anwendungen gehostet werden, identisch ist.

Die Standardeinstellung der Portnummer ist "8080".

Gültige Portnummern müssen im Bereich von 0 bis 65535 liegen.

Einstellungen der Richtlinie "Sicherheit"

Der Abschnitt "Sicherheit" enthält die Richtlinieneinstellung zum Konfigurieren der Sitzungsverschlüsselung und der Anmeldedatenverschlüsselung.

SecureICA-Mindestverschlüsselungsgrad

Mit dieser Einstellung geben Sie das Minimum für den Verschlüsselungsgrad der Sitzungsdaten an, die zwischen dem Server und einem Clientgerät ausgetauscht werden.

Wichtig:

Bei Virtual Delivery Agent 7.x kann mit dieser Richtlinieneinstellung nur die Anmeldedatenverschlüsselung mit RC5 128-Bit-Verschlüsselung aktiviert werden. Die anderen Einstellungen werden nur für Abwärtskompatibilität mit älteren Versionen von XenApp und XenDesktop bereitgestellt.

Bei VDA 7.x wird die Sitzungsdatenverschlüsselung mit den Grundeinstellungen der Bereitstellungsgruppe des VDAs festgelegt. Wenn für die Bereitstellungsgruppe die Option "Secure ICA aktivieren" ausgewählt ist, werden Sitzungsdaten mit der RC5-Verschlüsselung (128 Bit) verschlüsselt. Wenn die Option "Secure ICA aktivieren" für die Bereitstellungsgruppe nicht ausgewählt ist, werden Sitzungsdaten mit der Basic-Verschlüsselung verschlüsselt.

Wenn Sie diese Einstellung einer Richtlinie hinzufügen, wählen Sie eine der Optionen:

- Basic verschlüsselt die Clientverbindung mit einem nicht RC5-konformen Algorithmus. Mit diesem Verschlüsselungsverfahren kann der Datenstrom zwar vor direktem Lesen geschützt werden, ein Entschlüsseln ist aber möglich. Standardmäßig verwendet der Server für den Client-Server-Netzwerkverkehr den Verschlüsselungsgrad "Basic".
- RC5 (128 Bit) nur Anmeldung verschlüsselt die Anmeldedaten mit der RC5-128-Bit-Verschlüsselung und die Clientverbindung mit dem Verschlüsselungsgrad "Basic".
- RC5 (40 Bit) verschlüsselt die Verbindung mit der RC5-40-Bit-Verschlüsselung.
- RC5 (56 Bit) verschlüsselt die Verbindung mit der RC5-56-Bit-Verschlüsselung.
- RC5 (128 Bit) verschlüsselt die Verbindung mit der RC5-128-Bit-Verschlüsselung.

Die Einstellungen, die Sie für die Verschlüsselung zwischen Client und Server festlegen, können mit Verschlüsselungseinstellungen des Windows-Betriebssystems interagieren. Wenn ein höherer Verschlüsselungsgrad auf dem Server oder Benutzergerät eingestellt ist, können Einstellungen überschrieben werden, die Sie für veröffentlichte Ressourcen angegeben haben.

Sie können den Verschlüsselungsgrad erhöhen, um die Kommunikation und Datenintegrität für bestimmte Benutzer stärker zu sichern. Wenn für eine Richtlinie ein höherer Verschlüsselungsgrad erforderlich ist, wird Receivern mit einem niedrigeren Verschlüsselungsgrad die Verbindung verweigert.

SecureICA führt keine Authentifizierung durch und prüft auch nicht die Datenintegrität. Verwenden Sie SecureICA mit SSL/TLS-Verschlüsselung, um eine vollständige Verschlüsselung für die Site bereitzustellen.

SecureICA verwendet nicht FIPS-konforme Algorithmen. Wenn dies ein Problem ist, konfigurieren Sie den Server und die Receiver, um zu verhindern, dass SecureICA verwendet wird.

Einstellungen der Richtlinie "Serverlimits"

Der Abschnitt "Serverlimits" enthält Richtlinieneinstellungen, mit denen Sie Sitzungen im Leerlauf steuern.

Serverleerlauf-Zeitintervall

Mit dieser Einstellung geben Sie in Millisekunden an, wie lange eine ununterbrochene Benutzersitzung erhalten bleibt, wenn keine Benutzereingaben stattfinden.

Standardmäßig werden Leerlaufsitzungen nicht getrennt (Serverleerlaufzeitintervall = 0)

Einstellungen der Richtlinie "Sitzungslimits"

Der Abschnitt "Sitzungslimits" enthält Richtlinieneinstellungen, mit denen Sie die Dauer der Sitzungen festlegen können, bevor diese getrennt werden.

Timer für getrennte Sitzung

Mit dieser Einstellung aktivieren oder deaktivieren Sie einen Timer, der angibt, wie lange ein getrennter, gesperrter Desktop gesperrt bleibt, bevor die Sitzung abgemeldet wird.

Standardmäßig werden getrennte Sitzungen nicht abgemeldet.

Getrennte Sitzungen - Timerintervall

Mit dieser Einstellung legen Sie fest, wie viele Minuten ein getrennter, gesperrter Desktop gesperrt bleibt, bevor die Sitzung abgemeldet wird.

Standardmäßig sind es 1440 Minuten (24 Stunden).

Sitzungsverbindungstimer

Mit dieser Einstellung aktivieren oder deaktivieren Sie einen Timer, mit dem die maximale Dauer einer ununterbrochenen Sitzung zwischen einem Benutzergerät und einem Desktop festgelegt wird.

Standardmäßig ist dieser Timer aktiviert.

Sitzungsverbindung - Timerintervall

Mit dieser Einstellung legen Sie fest, wie viele Minuten eine ununterbrochene Verbindung zwischen einem Benutzergerät und einem Desktop maximal dauern darf.

Standardmäßig ist die maximale Dauer 1440 Minuten (24 Stunden).

Sitzungsleerlaufstimer

Mit dieser Einstellung aktivieren oder deaktivieren Sie einen Timer, der angibt, wie lange eine ununterbrochene Benutzergerätverbindung mit einem Desktop erhalten bleibt, wenn keine Benutzereingaben stattfinden.

Standardmäßig ist dieser Timer deaktiviert.

Sitzungsleerlauf - Timerintervall

Mit dieser Einstellung geben Sie in Minuten an, wie lange eine ununterbrochene Benutzergerätverbindung mit einem Desktop erhalten bleibt, wenn keine Benutzereingaben stattfinden.

Standardmäßig bleiben Leerlaufsitzen 1440 Minuten (24 Stunden) erhalten.

Einstellungen der Richtlinie "Sitzungszuverlässigkeit"

Der Abschnitt "Sitzungszuverlässigkeit" enthält Richtlinieneinstellungen zum Verwalten von Verbindungen, für die die Sitzungszuverlässigkeit verwendet wird.

Sitzungszuverlässigkeit - Verbindungen

Mit dieser Einstellung legen Sie fest, ob Sitzungen bei dem Verlust der Netzwerkkonnektivität offen bleiben sollen.

Standardmäßig ist die Sitzungszuverlässigkeit zugelassen.

Durch die Sitzungszuverlässigkeit bleiben Sitzungen aktiv und auf dem Bildschirm des Benutzers, wenn die Netzwerkverbindung unterbrochen wird. Die Benutzer sehen so lange weiterhin die Anwendung, die sie verwenden, bis die Netzwerkkonnektivität wiederhergestellt ist.

Mit der Sitzungszuverlässigkeit bleibt die Sitzung auf dem Server aktiv. Auf dem Client friert der Bildschirm ein und der Mauszeiger wird als Sanduhr angezeigt, bis die Verbindung am Ende des Tunnels wiederhergestellt ist. Der Benutzer kann während der Unterbrechung weiterhin auf die Anzeige zugreifen und mit der Anwendung weiterarbeiten, wenn die Netzwerkverbindung wiederhergestellt ist. Die Sitzungszuverlässigkeit verbindet Benutzer ohne Neuauthentifizierung wieder.

Wenn Sie möchten, dass Benutzer eine Verbindung zu unterbrochenen Sitzungen nur mit einer Neuauthentifizierung

wiederherstellen können, stellen Sie Authentifizierung bei automatischer Wiederverbindung von Clients entsprechend ein. Benutzer werden dann aufgefordert, sich neu zu authentifizieren, wenn sie sich mit der unterbrochenen Sitzung wiederverbinden.

Wenn Sie sowohl Sitzungszuverlässigkeit als auch die Funktion zur automatischen Wiederverbindung verwenden, werden beide Funktionen nacheinander ausgeführt. Die Sitzungszuverlässigkeit beendet oder trennt die Benutzersitzung, nachdem der mit der Option Sitzungszuverlässigkeit - Timeout festgelegte Zeitraum abgelaufen ist. Anschließend werden die für die automatische Wiederverbindung von Clients konfigurierten Einstellungen wirksam, und es wird versucht, eine Verbindung zur unterbrochenen Sitzung wiederherzustellen.

Sitzungszuverlässigkeit - Portnummer

Mit dieser Einstellung geben Sie die TCP-Portnummer für eingehende Sitzungszuverlässigkeitsverbindungen an.

Die Standardeinstellung der Portnummer ist "2598".

Sitzungszuverlässigkeit - Timeout

Mit dieser Einstellung legen Sie fest, wie viele Sekunden der Sitzungszuverlässigkeitsproxy auf die Wiederverbindung der Sitzung wartet, bevor die Sitzung getrennt wird.

Die Standardeinstellung ist 180 Sekunden (3 Minuten).

Sie können eine Sitzung länger offen lassen. Beachten Sie jedoch, dass Benutzer nicht zu einer Neuauthentifizierung aufgefordert werden, um den Bedienungskomfort zu erhöhen. Je länger eine Sitzung offen gelassen wird, desto höher ist das Risiko, dass der Benutzer abgelenkt wird und das Benutzergerät verlässt. Benutzer ohne Berechtigung hätten in dem Fall möglicherweise Zugriff auf die Sitzung.

Wenn Sie möchten, dass Benutzer eine Verbindung zu unterbrochenen Sitzungen nur mit einer Neuauthentifizierung wiederherstellen können, stellen Sie Authentifizierung bei automatischer Wiederverbindung von Clients entsprechend ein. Benutzer werden dann aufgefordert, sich neu zu authentifizieren, wenn sie sich mit der unterbrochenen Sitzung wiederverbinden.

Wenn Sie sowohl Sitzungszuverlässigkeit als auch die Funktion zur automatischen Wiederverbindung verwenden, werden beide Funktionen nacheinander ausgeführt. Die Sitzungszuverlässigkeit beendet oder trennt die Benutzersitzung, nachdem der mit der Option Sitzungszuverlässigkeit - Timeout festgelegte Zeitraum abgelaufen ist. Anschließend werden die für die automatische Wiederverbindung von Clients konfigurierten Einstellungen wirksam, und es wird versucht, eine Verbindung zur unterbrochenen Sitzung wiederherzustellen.

Einstellungen der Richtlinie "Zeitzonesteuerung"

Der Abschnitt "Zeitzonesteuerung" enthält Richtlinieneinstellungen für die Zeitzone in Sitzungen.

Lokale Zeitzone für Legacyclients schätzen

Mit dieser Einstellung aktivieren oder deaktivieren Sie das Schätzen der lokalen Zeitzone auf Clientgeräten, die falsche Zeitoneninformationen an den Server senden.

Standardmäßig schätzt der Server die lokale Zeitzone, wenn erforderlich.

Lokale Zeit des Clients verwenden

Mit dieser Einstellung legen Sie die Zeitzoneneinstellung der Benutzersitzung fest. Dies kann entweder die Zeitzone der Sitzung des Benutzers oder die des Benutzergeräts sein.

Standardmäßig wird die Zeitzone der Sitzung des Benutzers verwendet.

Damit diese Einstellung wirksam wird, aktivieren Sie die Einstellung Zeitzonenumleitung zulassen im Knoten "Remotedesktop-Sitzungshost" im Gruppenrichtlinien-Editor (Benutzerkonfiguration > Richtlinien > Administrative Vorlagen > Windows-Komponenten > Remotedesktopdienste > Remotedesktop-Sitzungshost > Geräte- und Ressourcenumleitung). Weitere Informationen über die Zeitzonenumleitung finden Sie im Citrix Knowledge Center.

Einstellungen der Richtlinie "TWAIN-Geräte"

Der Abschnitt "TWAIN-Geräte" enthält Richtlinieneinstellungen für die Zuordnung von TWAIN-Geräten, wie Digitalkameras oder Scanner, und für das Optimieren der Bildübertragung vom Server zum Client.

TWAIN-Geräteumleitung für Client

Mit dieser Einstellung legen Sie fest, ob Benutzer auf TWAIN-Geräte auf dem Benutzergerät aus Bildverarbeitungsanwendungen auf Servern zugreifen können. Standardmäßig ist die TWAIN-Geräteumleitung zugelassen.

Verwandte Richtlinieneinstellungen

- TWAIN-Komprimierungsgrad
- Bandbreitenlimit für TWAIN-Geräteumleitung
- Bandbreitenlimit für TWAIN-Geräteumleitung (Prozent)

TWAIN-Komprimierungsgrad

Mit dieser Einstellung geben Sie den Komprimierungsgrad für Bildübertragungen vom Client zum Server an. Verwenden Sie Gering für die beste Bildqualität, Mittel für eine gute Bilderqualität und Hoch für eine geringe Bildqualität. Standardmäßig wird die mittlere Komprimierung angewendet.

Einstellungen der Richtlinie "USB-Geräte"

Der Abschnitt "USB-Geräte" enthält Richtlinieneinstellungen für die Verwaltung der Dateiumleitung bei USB-Geräten.

Client-USB-Geräteumleitung

Mit dieser Einstellung legen Sie fest, ob die Umleitung von USB-Geräten zu und von Benutzergeräten zulässig ist.

Standardmäßig werden USB-Geräte nicht umgeleitet.

Regeln für die Client-USB-Geräteumleitung

Mit dieser Einstellung legen Sie die Umleitungsregeln für USB-Geräte fest.

In der Standardeinstellung sind keine Regeln angegeben.

Schließt ein Benutzer ein USB-Gerät an, prüft das Hostgerät jede Richtlinienregel, bis eine Übereinstimmung vorliegt. Die erste Übereinstimmung für ein beliebiges Gerät ist entscheidend. Ist es eine Zulassen-Regel, kann das Gerät remote auf dem virtuellen Desktop verwendet werden. Ist es eine Verweigern-Regel, kann das Gerät nur auf dem lokalen Desktop

verwendet werden. Wenn keine Übereinstimmung gefunden wird, werden die Standardregeln verwendet.

Richtlinienregeln haben das Format {Allow:|Deny:} plus Tag=Wert, durch Leerzeichen getrennt. Die folgenden Tags werden unterstützt:

Tagname	Beschreibung
VID	Vendor-ID vom Gerätedeskriptor
PID	Produkt-ID vom Gerätedeskriptor
REL	Release-ID vom Gerätedeskriptor
Class	Klasse vom Gerätedeskriptor oder ein Schnittstellendeskriptor
SubClass	Unterklasse vom Gerätedeskriptor oder ein Schnittstellendeskriptor
Prot	Protokoll vom Gerätedeskriptor oder ein Schnittstellendeskriptor

Wenn Sie eine neue Richtlinienregel erstellen, beachten Sie Folgendes:

- Bei Regeln wird die Groß- und Kleinschreibung nicht berücksichtigt.
- Regeln können optional von einem Kommentar gefolgt werden, der mit # eingeleitet wird.
- Leere Zeilen und Kommentare werden ignoriert.
- Tags müssen den Übereinstimmungsoperator = verwenden. Beispielsweise VID=1230.
- Jede Regel muss auf einer neuen Zeile beginnen oder Teil einer durch Semikolon getrennten Liste sein.
- USB-Klassencodes finden Sie auf der Website von USB Implementers Forum, Inc.

Beispiel für administratordefinierte USB-Richtlinienregeln

Allow: VID=1230 PID=0007 # Weitere Industrie, Weiteres Flash-Laufwerk

Deny: Class=08 SubClass=05 # Massenspeichergeräte

Eine Regel, die alle USB-Geräte verweigert, erstellen Sie mit "DENY:" ohne weitere Tags.

Client-USB-Geräteumleitung für Plug & Play-Geräte

Mit dieser Einstellung legen Sie fest, ob Plug & Play-Geräte, wie Kameras oder POS-Geräte (Point of Sale) in einer Clientsitzung verwendet werden können.

In der Standardeinstellung ist die Umleitung von Plug & Play-Geräten zugelassen. Bei der Einstellung Zugelassen werden alle Plug & Play-Geräte für einen bestimmten Benutzer oder eine bestimmte Benutzergruppe umgeleitet. Bei der Einstellung Nicht zugelassen werden keine Geräte umgeleitet.

Einstellungen der Richtlinie "Visuelle Anzeige"

Der Abschnitt "Visuelle Anzeige" enthält Richtlinieneinstellungen, mit denen die Qualität der von virtuellen Desktops an das Benutzergerät gesendeten Bilder gesteuert wird.

Frateratesollwert

Mit dieser Einstellung geben Sie die maximale Anzahl von Frames pro Sekunde an, die vom virtuellen Desktop zum Benutzergerät gesendet werden.

In der Standardeinstellung ist die Höchstanzahl 30 Frames pro Sekunde.

Die Festlegung auf eine hohe Anzahl von Frames pro Sekunde (z. B. 30) führt zu einer besseren Benutzererfahrung, erfordert aber mehr Bandbreite. Wenn Sie die Anzahl von Frames pro Sekunde herabsetzen (z. B. auf 10), wird die Serverskalierbarkeit auf Kosten der Benutzererfahrung erhöht.

Bildqualität

Mit dieser Einstellung legen Sie die Bildqualität für auf dem Benutzergerät angezeigte Bilder fest.

Die Standardeinstellung ist "Mittel".

Zum Festlegen der Bildqualität wählen Sie eine der folgenden Optionen:

- Niedrig
- Mittel
- Hoch
- Zu verlustfrei verbessern
- Immer verlustfrei

In Situationen, in denen kein Qualitätsverlust akzeptabel ist (z. B. bei der Anzeige von Röntgenbildern), wählen Sie Immer verlustfrei, um sicherzustellen, dass keine verlustreichen Daten an das Benutzergerät gesendet werden.

Bei Auswahl von Zu verlustfrei verbessern werden verlustreiche Bilder in Zeiträumen mit hoher Netzwerkaktivität und verlustfreie Bilder bei verringerter Netzwerkaktivität gesendet.

Hinweis: Wenn die Einstellung Legacygrafikmodus aktiviert ist, hat die Einstellung Bildqualität der Richtlinie keine Auswirkungen.

Einstellungen der Richtlinie "Bewegtbilder"

Der Abschnitt "Bewegtbilder" enthält Einstellungen, mit denen Sie die Komprimierung für dynamische Bilder entfernen oder ändern können.

Mindestbildqualität

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung wird die zulässige Mindestbildqualität für den adaptiven Bildschirm angegeben. Je geringer die verwendete Komprimierung ist, desto höher ist die Qualität der angezeigten Bilder. Es stehen folgende Komprimierungen zur Verfügung: Ultrahoch, Sehr hoch, Hoch, Normal und Niedrig.

Die Standardeinstellung ist "Normal".

Bewegtbildkomprimierung

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung wird angegeben, ob der adaptive Bildschirm aktiviert ist. Der adaptive Bildschirm passt die Bildqualität von Videos und Bildübergängen in Diashows auf der Grundlage der verfügbaren Bandbreite automatisch an. Bei aktiviertem adaptivem Bildschirm werden Benutzern gleichmäßig ausgeführte Präsentationen ohne Qualitätseinbußen angezeigt.

Standardmäßig ist der adaptive Bildschirm aktiviert.

Grad der progressiven Komprimierung

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung wird zuerst ein weniger detailliertes Bild angezeigt, das dafür aber schneller dargestellt werden kann.

In der Standardeinstellung wird keine progressive Komprimierung angewendet.

Sobald es verfügbar ist, wird ein detailreicheres Bild angezeigt, das die normale Einstellung für verlustreiche Komprimierung verwendet. Verwenden Sie sehr hohe oder ultrahohe Komprimierung für die verbesserte Anzeige von bandbreitenintensiven Grafiken, wie etwa Fotografien.

Die progressive Komprimierung ist nur wirksam, wenn der Komprimierungsgrad höher ist als die Einstellung für Grad der verlustreichen Komprimierung.

Hinweis: Der stärkere Komprimierungsgrad für die progressive Komprimierung verbessert auch die Interaktivität von dynamischen Bildern über Clientverbindungen. Die Qualität eines dynamischen Bilds, z. B. ein sich drehendes dreidimensionales Modell, wird temporär verringert, bis das Bild stehen bleibt. Zu dem Zeitpunkt wird dann die reguläre Einstellung der verlustreichen Komprimierung angewendet.

Verwandte Richtlinieneinstellungen

- Schwellenwert für progressive Komprimierung
- Progressive Heavyweight-Komprimierung

Schwellenwert für progressive Komprimierung

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung geben Sie die maximale Bandbreite in Kilobits pro Sekunde für eine Verbindung an, auf die progressive Komprimierung angewendet wird. Die Komprimierung wird nur für Clientverbindungen unter diesem Bandbreitenwert verwendet.

Der Standardschwellenwert ist 2147483647 Kilobits pro Sekunde.

Verwandte Richtlinieneinstellungen

- Schwellenwert für progressive Komprimierung
- Progressive Heavyweight-Komprimierung

Mindestframeratesollwert

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung wird die Framerate pro Sekunde eingestellt, die das System für dynamische Bilder in Netzwerken mit geringer Bandbreite versucht beizubehalten.

Die Standardeinstellung für diesen Parameter ist 10 F/s.

Einstellungen der Richtlinie "Standbilder"

Der Abschnitt "Standbilder" enthält Einstellungen, mit denen Sie die Komprimierung für statische Bilder entfernen oder ändern können.

Zusätzliche Farbkomprimierung

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung aktivieren oder deaktivieren Sie die Verwendung der zusätzlichen Farbkomprimierung für Bilder, die über Clientverbindungen mit beschränkter Bandbreite bereitgestellt werden; dies verbessert die Reaktionszeit, da die Bilder in geringerer Qualität angezeigt werden.

Standardmäßig ist die zusätzliche Farbkomprimierung deaktiviert.

Bei Aktivierung wird die zusätzliche Farbkomprimierung nur angewendet, wenn die Bandbreite der Clientverbindung unter dem für Schwellenwert für zusätzliche Farbkomprimierung festgelegten Wert liegt. Wenn die Bandbreite der Clientverbindung über dem Schwellenwert liegt oder Deaktiviert ausgewählt ist, wird die zusätzliche Farbkomprimierung nicht angewendet.

Schwellenwert für zusätzliche Farbkomprimierung

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung geben Sie die maximale Bandbreite in Kilobits pro Sekunde für eine Verbindung an, unter der die zusätzliche Farbkomprimierung angewendet wird. Wenn die Bandbreite der Clientverbindung unter den eingestellten Wert abfällt, wird die zusätzliche Farbkomprimierung (falls aktiviert) angewendet.

Der Standardschwellenwert ist 8192 Kilobits pro Sekunde.

Heavyweight-Komprimierung

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung reduzieren Sie die erforderliche Bandbreite noch stärker als mit der progressiven Komprimierung, ohne dabei an Bildqualität zu verlieren, indem ein verbesserter grafischer Algorithmus verwendet wird, der aber mehr CPU beansprucht.

Standardmäßig ist die Heavyweight-Komprimierung deaktiviert.

Wenn die Heavyweight-Komprimierung aktiviert ist, gilt sie für alle verlustreichen Komprimierungen. Diese Einstellung wird von Citrix Receiver unterstützt, hat aber keine Auswirkung auf andere Plug-Ins.

Verwandte Richtlinieneinstellungen

- Grad der progressiven Komprimierung
- Schwellenwert für progressive Komprimierung

Grad der verlustreichen Komprimierung

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus

aktiviert ist.

Mit dieser Einstellung steuern Sie den Grad der verlustreichen Komprimierung, der für Grafiken verwendet wird, die über Clientverbindungen mit beschränkter Bandbreite bereitgestellt werden. In solchen Fällen kann die Anzeige von Bildern ohne Komprimierung sehr langsam sein.

Standardmäßig wird eine mittlere Komprimierung ausgewählt.

Bessere Reaktionszeiten bei bandbreitenintensiven Bildern erzielen Sie mit hoher Komprimierung. In Fällen, in denen die Bilddaten erhalten bleiben müssen, beispielsweise bei der Anzeige von Röntgenbildern, wo kein Qualitätsverlust akzeptabel ist, sollten Sie die verlustreiche Komprimierung nicht einsetzen.

Verwandte Richtlinieneinstellungen

- Schwellenwert für verlustreiche Komprimierung

Schwellenwert für verlustreiche Komprimierung

Hinweis: Bei Virtual Delivery Agent 7.x gilt diese Richtlinieneinstellung nur, wenn die Richtlinieneinstellung Legacygrafikmodus aktiviert ist.

Mit dieser Einstellung geben Sie die maximale Bandbreite in Kilobits pro Sekunde für eine Verbindung an, auf die die verlustreiche Komprimierung angewendet wird.

Der Standardschwellenwert ist 2147483647 Kilobits pro Sekunde.

Wenn Sie die Einstellung Grad der verlustreichen Komprimierung einer Richtlinie hinzufügen, ohne einen Schwellenwert anzugeben, kann sich dadurch die Anzeigegeschwindigkeit für detailreiche Bitmaps, wie Fotografien, über ein LAN verbessern.

Verwandte Richtlinieneinstellungen

- Grad der verlustreichen Komprimierung

Einstellungen der Richtlinie "WebSockets"

Der Abschnitt "WebSockets" enthält Richtlinieneinstellungen für den Zugriff auf virtuelle Desktops und gehostete Anwendungen mit Receiver für HTML5. Das Feature WebSockets erhöht die Sicherheit und verringert die Last durch bidirektionale Kommunikation zwischen browserbasierten Anwendungen und Servern ohne Öffnen von mehreren HTTP-Verbindungen.

WebSockets-Verbindungen

Diese Einstellung lässt WebSockets-Verbindungen zu oder lehnt sie ab.

Standardmäßig sind WebSocket-Verbindungen nicht zulässig.

WebSockets-Portnummer

Mit dieser Einstellung wird der Port für eingehende WebSocket-Verbindungen festgelegt.

In der Standardeinstellung ist der Wert 8008.

Vertrauenswürdige WebSockets-Ursprungsserverliste

Diese Einstellung bietet eine durch Trennzeichen getrennte Liste der vertrauenswürdigen Ursprungsserver, normalerweise Receiver für Web, in Form von URLs. Nur WebSockets-Verbindungen, die von einer dieser Adressen stammen, werden vom Server akzeptiert.

Standardmäßig wird der Platzhalter * verwendet. Damit wird allen Receiver für Web-URLs vertraut.

Wenn Sie eine Adresse in die Liste eingeben möchten, verwenden Sie folgende Syntax:

<Protokoll>://<Vollqualifizierter Domänenname des Hosts>:[Port]

Das Protokoll muss HTTP oder HTTPS sein. Wenn der Port nicht angegeben wird, wird Port 80 für HTTP und Port 443 für HTTPS verwendet.

Der Platzhalter * kann innerhalb der URL verwendet werden, außer als Teil einer IP-Adresse (10.105. *. *).

Einstellungen der Richtlinie "Lastverwaltung"

Sep 29, 2015

Der Abschnitt "Lastverwaltung" enthält Richtlinieneinstellungen für das Aktivieren und Konfigurieren des Lastausgleichs zwischen Servern, über die Windows Server-Betriebssystemmaschinen bereitgestellt werden.

Toleranzwert für gleichzeitige Anmeldungen

Mit dieser Einstellung geben Sie die maximal zulässige Anzahl gleichzeitiger Anmeldungen bei einem Server an.

Die Standardeinstellung ist 2.

CPU-Nutzung

Mit dieser Einstellung geben Sie den Prozentsatz der CPU-Nutzung an, bei dem der Server Volllast meldet. Ist diese Einstellung aktiviert, beträgt der Standardwert, bei dem der Server Volllast meldet, 90 %.

Standardmäßig ist diese Einstellung deaktiviert und die CPU-Nutzung wird bei der Lastberechnung nicht berücksichtigt.

CPU-Nutzung ausschließlich Prozesspriorität

Mit dieser Einstellung geben Sie die Prioritätsstufe an, bei der die Prozess-CPU-Auslastung vom Lastindex der CPU-Nutzung ausgeschlossen wird.

Die Standardeinstellung ist "Unter normal oder niedrig".

Datenträgernutzung

Mit dieser Einstellung geben Sie die Länge der Datenträgerwarteschlange an, zu der der Server 75 % Volllast meldet. Der Standardwert dieser Einstellung ist 8.

Standardmäßig ist diese Einstellung deaktiviert und die Datenträgernutzung wird bei der Lastberechnung nicht berücksichtigt.

Sitzungshöchstanzahl

Mit dieser Einstellung geben Sie die maximale Anzahl von Sitzungen an, die von einem Server gehostet werden können. Ist die Einstellung aktiviert, beträgt der Standardwert für die maximale Anzahl Sitzungen an, die von einem Server gehostet werden können, 250.

Standardmäßig ist diese Einstellung aktiviert.

Speichernutzung

Mit dieser Einstellung geben Sie den Prozentsatz der Speichernutzung an, bei dem der Server Volllast meldet. Ist diese Einstellung aktiviert, beträgt der Standardwert, bei dem der Server Volllast meldet, 90 %.

Standardmäßig ist diese Einstellung deaktiviert und die Speichernutzung wird bei der Lastberechnung nicht berücksichtigt.

Speichernutzung - Ausgangslast

Mit dieser Einstellung geben Sie einen Näherungswert der Speichernutzung durch das Basisbetriebssystem in MB an, unterhalb dessen die Speichernutzung bei einem Server als Nulllast interpretiert wird.

Der Standardwert dieser Einstellung ist 768 MB.

Einstellungen der Richtlinie "Profilverwaltung"

Sep 29, 2015

Der Abschnitt "Profilverwaltung" enthält Richtlinieneinstellungen zum Aktivieren der Profilverwaltung und zum Konfigurieren der Gruppen, die in die Verarbeitung der Profilverwaltung eingeschlossen bzw. ausgeschlossen werden sollen.

Weitere Informationen, wie z. B. die Namen der entsprechenden INI-Dateieinstellungen und die für eine bestimmte Richtlinieneinstellung erforderliche Version der Profilverwaltung finden Sie in der [Referenz für die Profilverwaltungs-ADM-Datei](#).

Einstellungen der Richtlinie "Erweitert"

Der Abschnitt "Erweitert" enthält Richtlinieneinstellungen für die erweiterte Konfiguration der Profilverwaltung.

Automatische Konfiguration deaktivieren

Mit dieser Einstellung legen Sie fest, dass die Umgebung von der Profilverwaltung untersucht werden kann, z. B., um zu überprüfen, ob persönliche vDisks vorhanden sind, und um die Gruppenrichtlinie entsprechend zu konfigurieren. Nur Richtlinien für die Profilverwaltung mit dem Status "Nicht konfiguriert" werden angepasst, sodass alle zuvor vorgenommenen Anpassungen beibehalten werden. Dieses Feature beschleunigt die Bereitstellung und vereinfacht die Optimierung. Es ist keine Konfiguration des Features erforderlich, aber Sie können die automatische Konfiguration bei Upgrades (zum Beibehalten der Einstellungen von früheren Versionen) oder bei der Problembehandlung deaktivieren. Die automatische Konfiguration funktioniert in XenApp oder anderen Umgebungen nicht.

In der Standardeinstellung ist die automatische Konfiguration zugelassen.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, wird die automatische Konfiguration aktiviert. Einstellungen der Profilverwaltung können sich dann möglicherweise ändern, wenn sich die Umgebung ändert.

Benutzer bei Problem abmelden

Mit dieser Einstellung wird eine Abmeldung eines Benutzers durch die Profilverwaltung ermöglicht, wenn ein Problem auftritt, z. B. wenn der Benutzerspeicher nicht verfügbar ist. Wenn diese Einstellung aktiviert ist, wird dem Benutzer eine Fehlermeldung angezeigt, bevor er abgemeldet wird. Wenn diese Einstellung deaktiviert ist, erhalten Benutzer ein temporäres Profil.

Standardmäßig ist diese Einstellung deaktiviert und Benutzer erhalten ein temporäres Profil, wenn ein Problem auftritt.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, wird ein temporäres Profil bereitgestellt.

Anzahl Wiederholungen beim Zugriff auf gesperrte Dateien

Mit dieser Einstellung geben Sie die Anzahl der Zugriffsversuche auf gesperrte Dateien durch die Profilverwaltung an.

Der Standardwert dieser Einstellung ist fünf Versuche.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, wird der Standardwert verwendet.

Internet-Cookiedateien bei Abmeldung verarbeiten

Mit dieser Einstellung legen Sie fest, dass die Datei index.dat beim Abmelden von der Profilverwaltung verarbeitet werden kann, damit Internet-Cookies im Dateisystem nach längerem Browsen entfernt werden. Auf diese Weise wird eine Aufblähung von Profilen vermieden. Eine Aktivierung der Einstellung verlängert die Abmeldezeiten. Aktivieren Sie sie daher nur, wenn das entsprechende Problem bei Ihnen auftritt.

Standardmäßig ist diese Einstellung deaktiviert und die Profilverwaltung verarbeitet die Datei index.dat beim Abmelden nicht.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, wird Index.dat nicht verarbeitet.

Grundlegende Richtlinieneinstellungen

Der Abschnitt "Grundlegend" enthält Richtlinieneinstellungen für die grundlegende Konfiguration der Profilverwaltung.

Aktiv zurückschreiben

Mit dieser Einstellung können geänderte Dateien und Ordner (aber keine Registrierungseinträge) während einer Sitzung mit dem Benutzerspeicher synchronisiert werden bevor die Abmeldung erfolgt.

Standardmäßig ist die Synchronisierung mit dem Benutzerspeicher während laufender Sitzungen deaktiviert.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, ist sie aktiviert.

Profilverwaltung aktivieren

Mit dieser Einstellung wird die Profilverwaltung zur Verarbeitung von An- und Abmeldungen aktiviert.

Standardmäßig ist diese Einstellung deaktiviert, um die Bereitstellung zu vereinfachen.

Wichtig: Citrix empfiehlt, eine Aktivierung der Profilverwaltung erst dann durchzuführen, wenn alle anderen Setupaufgaben ausgeführt wurden und getestet wurde, wie sich Citrix Benutzerprofile in der Umgebung verhalten.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, verarbeitet die Profilverwaltung keine Windows-Benutzerprofile.

Ausgeschlossene Gruppen

Mit dieser Einstellung geben Sie an, welche lokalen Computergruppen und welche Domänengruppen (lokale, globale und universelle) nicht von der Profilverwaltung verarbeitet werden sollen.

Wenn diese Option aktiviert ist, verarbeitet die Profilverwaltung keine Mitglieder der angegebenen Benutzergruppen.

Standardmäßig ist diese Einstellung deaktiviert und Mitglieder aller Benutzergruppen werden verarbeitet.

Geben Sie Domänengruppen im Format \ an.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Mitglieder aller Benutzergruppen verarbeitet.

Unterstützung von Offlineprofilen

Mit dieser Einstellung wird die Unterstützung von Offlineprofilen aktiviert, durch die Profile nach einer Unterbrechung der Verbindung mit dem Netzwerk zum frühestmöglichen Zeitpunkt mit dem Benutzerspeicher synchronisiert werden.

Standardmäßig ist die Unterstützung von Offlineprofilen deaktiviert.

Diese Einstellung gilt für Benutzer von Laptops und mobile Benutzer im Roamingmodus. Wenn die Verbindung zum Netzwerk unterbrochen wird, bleiben die Profile auf dem Laptop oder Gerät intakt, selbst wenn das Gerät neu gestartet wird oder im Ruhezustand gewesen ist. Während mobile Benutzer arbeiten, werden ihre Profile lokal aktualisiert und am Ende mit dem Benutzerspeicher synchronisiert, wenn die Netzwerkverbindung wiederhergestellt ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, ist die Unterstützung von Offlineprofilen deaktiviert.

Pfad zum Benutzerspeicher

Mit dieser Einstellung geben Sie den Pfad zu dem Verzeichnis (Benutzerspeicher) an, in dem Benutzereinstellungen, z. B. Registrierungseinstellungen und synchronisierte Dateien, gespeichert werden.

Standardmäßig wird das Windows-Verzeichnis auf dem Basislaufwerk verwendet.

Wenn diese Einstellung deaktiviert ist, werden die Benutzereinstellungen im Windows-Unterverzeichnis des Basisverzeichnisses gespeichert.

Mögliche Pfade:

- **Relativer Pfad:** Dieser Pfad muss relativ zum Basisverzeichnis sein. Dieses ist normalerweise mit dem Attribut #homeDirectory# für einen Benutzer in Active Directory konfiguriert.
- **Absoluter UNC-Pfad:** Hiermit wird üblicherweise eine Serverfreigabe oder ein DFS-Namespace angegeben.
- **Deaktiviert oder nicht konfiguriert:** In diesem Fall wird als Wert #homeDirectory#\Windows angenommen.

Verwenden Sie die folgenden Variablentypen beim Konfigurieren dieser Richtlinieneinstellung:

- Systemumgebungsvariablen in Prozentzeichen (z. B. %ProfVer%). Beachten Sie, dass Systemumgebungsvariablen normalerweise weitere Einrichtung erfordern.
- Attribute des Active Directory-Benutzerobjekts in Rauten (z. B. #sAMAccountName#).
- Profilverwaltungsvariablen. Weitere Informationen finden Sie unter "Profilverwaltung" in eDocs.

Sie können auch mit den Benutzerumgebungsvariablen %username% und %userdomain% benutzerdefinierte Attribute erstellen, um Organisationsvariablen wie Standort und Benutzer vollständig zu definieren. Bei Attributen muss Groß- und Kleinschreibung beachtet werden.

Beispiele:

- \\Server\Freigabe#\#sAMAccountName# speichert die Benutzereinstellungen unter dem UNC-Pfad

- \\Server\Freigabe\KurtMeier (wenn #sAMAccountName# zu Kurt Meier als aktuellem Benutzer aufgelöst wird).
- \\server\profiles\$\%USERNAME%.%USERDOMAIN%\!CTX_PROFILEVER!!CTX_OSBITNESS! kann erweitert werden zu \\server\profiles\$\JohnSmith.DOMAINCONTROLLER1\v2x64

Wichtig: Unabhängig davon, welche Attribute oder Variablen Sie verwenden, müssen Sie sicherstellen, dass diese Einstellung zu einem Ordner über dem Ordner, der NTUSER.DAT enthält, aufgelöst wird. Wenn sich diese Datei z. B. in \\Server\profiles\$\KurtMeier.BuHa\v2x64\UPM_Profile befindet, geben Sie den Pfad zum Benutzerspeicher als \\Server\profiles\$\KurtMeier.BuHa\v2x64 an (ohne den Unterordner \UPM_Profile) an. Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, wird das Windows-Verzeichnis auf dem Basislaufwerk verwendet.

Anmeldungen lokaler Administratoren verarbeiten

Mit dieser Einstellung wird angegeben, ob Anmeldungen von Mitgliedern der Gruppe "BUILTIN\Administrators" verarbeitet werden. Dies ermöglicht es Domänenbenutzern mit lokalen Administratorrechten, d. h. normalerweise Benutzer mit zugewiesenen virtuellen Desktops, die Verarbeitung zu umgehen, sich anzumelden und Probleme mit der Profilverwaltung bei einem Desktop zu behandeln.

Wenn diese Einstellung unter Serverbetriebssystemen deaktiviert oder nicht konfiguriert ist, nimmt die Profilverwaltung an, dass Anmeldungen von Domänenbenutzern, aber nicht von lokalen Administratoren, verarbeitet werden müssen. Unter Desktopbetriebssystemen werden Anmeldungen lokaler Administratoren verarbeitet.

Diese Einstellung ist standardmäßig deaktiviert und Anmeldungen lokaler Administratoren werden nicht verarbeitet.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Anmeldungen lokaler Administratoren nicht verarbeitet.

Verarbeitete Gruppen

Mit dieser Einstellung geben Sie an, welche lokalen Computergruppen und welche Domänengruppen (lokale, globale und universelle) von der Profilverwaltung verarbeitet werden sollen.

Wenn diese Option aktiviert ist, verarbeitet die Profilverwaltung nur Mitglieder der angegebenen Benutzergruppen.

Standardmäßig ist diese Einstellung deaktiviert und Mitglieder aller Benutzergruppen werden verarbeitet.

Geben Sie Domänengruppen im Format \ an.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Mitglieder aller Benutzergruppen verarbeitet.

Einstellungen der Richtlinie "Cross-Platform"

Der Abschnitt "Plattformübergreifende Einstellungen" enthält Richtlinieneinstellungen für die Konfiguration der plattformübergreifenden Einstellungen der Profilverwaltung.

Benutzergruppen für plattformübergreifende Einstellungen

Mit dieser Einstellung geben Sie die Benutzergruppen an, deren Profile verarbeitet werden sollen, wenn das Feature für plattformübergreifende Einstellungen aktiviert ist.

Standardmäßig ist diese Einstellung deaktiviert und alle Benutzergruppen in der Richtlinieneinstellung Verarbeitete Gruppen werden verarbeitet.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden alle Benutzergruppen verarbeitet.

Plattformübergreifende Einstellungen aktivieren

Mit dieser Einstellung aktivieren oder deaktivieren Sie das Feature "Plattformübergreifende Einstellungen". Dieses ermöglicht das Migrieren und Roamen von Benutzerprofilen, wenn ein Benutzer eine Verbindung mit einer Anwendung herstellt, die unter mehreren Betriebssystemen ausgeführt wird.

Standardmäßig ist das Feature "Plattformübergreifende Einstellungen" deaktiviert.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden keine plattformübergreifenden Einstellungen angewendet.

Pfad zu plattformübergreifenden Definitionen

Mit dieser Einstellung geben Sie in Form eines UNC-Pfads den Netzwerkspeicherort an, der die aus dem Downloadpaket kopierten Definitionsdateien enthält.

Hinweis: Benutzer benötigen Lesezugriff und Administratoren Schreibzugriff auf diesen Speicherort, bei dem es sich entweder um eine Server Message Block-Dateifreigabe (SMB) oder eine Common Internet File System-Dateifreigabe (CIFS) handeln muss.

In der Standardeinstellung ist kein Pfad angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden keine plattformübergreifenden Einstellungen angewendet.

Pfad zum Speicher für plattformübergreifende Einstellungen

Diese Einstellung gibt den Pfad zum Speicher für plattformübergreifende Einstellungen an. Hier werden die plattformübergreifenden Einstellungen der Benutzer gespeichert. Der Pfad kann ein UNC-Pfad oder ein relativer Pfad zum Basisverzeichnis sein.

Hinweis: Benutzer müssen über Schreibzugriff auf den Speicher für plattformübergreifende Einstellungen verfügen. Standardmäßig ist diese Einstellung deaktiviert und der Pfad "Windows\PM_CP" wird verwendet.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, wird der Standardwert verwendet.

Quelle für plattformübergreifende Einstellungen

Mit dieser Einstellung legen Sie eine Plattform als

— *Basisplattform*

fest, wenn die entsprechende Einstellung für die Organisationseinheit der Plattform aktiviert ist. Daten von den Profilen der Basisplattform werden in den Speicher für plattformübergreifende Einstellungen migriert.

Die Profile jeder Plattform werden in einer separaten Organisationseinheit gespeichert. Dies bedeutet, dass Sie die Plattform auswählen müssen, deren Profildaten zum Füllen des Speichers für plattformübergreifende Einstellungen verwendet werden sollen. Dies wird als

— *Basisplattform*

bezeichnet.

Wenn diese Option aktiviert ist, migriert die Profilverwaltung die Daten des Einzelplattformprofils in den Speicher, wenn der Speicher für plattformübergreifende Einstellungen eine Definitionsdatei ohne Daten enthält oder die zwischengespeicherten Daten eines Einzelplattformprofils neuer sind als die Definitionsdaten im Speicher.

Wichtig: Wenn diese Einstellung in mehreren Organisationseinheiten für mehrere Benutzer oder Maschinenobjekte aktiviert ist, wird die Plattform, bei der sich der erste Benutzer anmeldet, zum Basisprofil.

Standardmäßig ist diese Einstellung deaktiviert und die Profilverwaltung migriert die Daten des Einzelplattformprofils nicht in den Speicher.

Einstellungen der Ausschlussrichtlinie für das Dateisystem

Der Abschnitt "Ausschlüsse" enthält Richtlinieneinstellungen zum Konfigurieren der Dateien und Verzeichnisse in einem Benutzerprofil, die von der Synchronisierung ausgeschlossen werden sollen.

Ausschlussliste - Verzeichnisse

Mit dieser Einstellung geben Sie eine Liste der Ordner im Benutzerprofil an, die bei der Synchronisierung ignoriert werden sollen.

Geben Sie die Ordernamen als Pfad relativ zum Benutzerprofil (% USERPROFILE%) an.

Standardmäßig ist diese Einstellung deaktiviert und alle Ordner in Benutzerprofilen werden synchronisiert.

Beispiele:

- Desktop ignoriert den Ordner "Desktop" im Benutzerprofil.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden alle Ordner im Benutzerprofil synchronisiert.

Ausschlussliste - Dateien

Mit dieser Einstellung geben Sie eine Liste der Dateien im Benutzerprofil an, die bei der Synchronisierung ignoriert werden sollen.

Standardmäßig ist diese Einstellung deaktiviert und alle Dateien in Benutzerprofilen werden synchronisiert.

Geben Sie die Dateinamen als Pfad relativ zum Benutzerprofil (% USERPROFILE%) an. Hinweis: Platzhalter sind zulässig und werden rekursiv angewendet.

Beispiele:

- "Desktop\Desktop.ini" ignoriert die Datei Desktop.ini im Verzeichnis "Desktop".

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden alle Dateien im Benutzerprofil synchronisiert.

Einstellungen der Synchronisierungsrichtlinie für das Dateisystem

Aktualisiert: 2014-05-02

Der Abschnitt "Synchronisierung" enthält Richtlinieneinstellungen für die Angabe, welche Dateien und Ordner in einem Benutzerprofil zwischen dem System, auf dem das Profil installiert ist, und dem Benutzerspeicher synchronisiert werden.

Zu synchronisierende Verzeichnisse

Diese Einstellung gibt die Dateien an, die sich in ausgeschlossenen Ordnern befinden und von der Profilverwaltung in die Synchronisierung eingeschlossen werden sollen. Standardmäßig synchronisiert die Profilverwaltung alle Elemente im Benutzerprofil. Es ist nicht erforderlich, Unterordner des Benutzerprofils einzuschließen, indem Sie sie dieser Liste hinzufügen. Weitere Informationen finden Sie unter [Aufnehmen und Ausschließen von Objekten](#).

Pfade in dieser Liste müssen relativ zum Benutzerprofil sein.

Beispiele:

- Desktop\ausschließen\aufnehmen gibt den Unterordner "aufnehmen" des Verzeichnisses Desktop\ausschließen an.

Standardmäßig ist diese Einstellung deaktiviert und keine Ordner sind angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden nur nicht ausgeschlossene Ordner im Benutzerprofil synchronisiert.

Zu synchronisierende Dateien

Diese Einstellung gibt die Dateien an, die sich in ausgeschlossenen Ordnern befinden und von der Profilverwaltung in die Synchronisierung eingeschlossen werden sollen. Standardmäßig synchronisiert die Profilverwaltung alle Elemente im Benutzerprofil. Es ist nicht erforderlich, Dateien in dem Benutzerprofil einzuschließen, indem Sie sie dieser Liste hinzufügen. Weitere Informationen finden Sie unter [Aufnehmen und Ausschließen von Objekten](#).

Pfade in dieser Liste müssen relativ zum Benutzerprofil sein. Relative Pfade werden als relativ zum Benutzerprofil interpretiert. Platzhalter können nur für Dateinamen verwendet werden. Platzhalter können nicht verschachtelt werden und werden rekursiv angewendet.

Beispiele:

- Anwendungsdaten\Lokal\Microsoft\Office\Access.qat gibt eine Datei unter einem Ordner an, der in der Standardkonfiguration ausgeschlossen wurde.
- Anwendungsdaten\Lokal\Anwendungen*.cfg gibt alle Dateien mit der Erweiterung .cfg im Profilordner Anwendungsdaten\Lokal\Anwendungen und dessen Unterordnern an.

Standardmäßig ist diese Einstellung deaktiviert und keine Dateien sind angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden nur nicht ausgeschlossene Dateien im Benutzerprofil synchronisiert.

Zu spiegelnde Ordner

Mit dieser Einstellung geben Sie an, welche Ordner relativ zum Stammordner eines Benutzerprofils gespiegelt werden sollen. Diese Richtlinie kann beim Lösen von Problemen bei Transaktionsordnern (auch "Referenzordner" genannt) helfen. Dies sind Ordner, die voneinander abhängige Dateien enthalten, wobei eine Datei Verweise auf andere Dateien enthält.

Durch das Spiegeln von Ordnern kann die Profilverwaltung einen Transaktionsordner und seinen Inhalt als eine Entität verarbeiten. So wird das Aufblähen von Profilen verhindert. Beachten Sie, dass in solchen Situationen der "letzte Schreibvorgang Priorität hat", d. h. dass Dateien in gespiegelten Ordnern, die in mehr als einer Sitzung geändert wurden, von der letzten Aktualisierung überschrieben werden. Hierdurch gehen Profiländerungen verloren.

Sie können z. B. den Cookies-Ordner von Internet Explorer spiegeln, damit Index.dat mit den Cookies synchronisiert wird, auf die die Datei verweist.

Wenn ein Benutzer zwei Internet Explorer-Sitzungen auf unterschiedlichen Servern hat und er in jeder Sitzung auf andere Websites zugreift, werden Cookies dieser Sites auf dem entsprechenden Server hinzugefügt. Wenn sich der Benutzer von der ersten Sitzung abmeldet (oder auch mitten in der Sitzung, wenn das Feature für aktives Zurückschreiben konfiguriert ist), sollten die Cookies der zweiten Sitzung die Cookies der ersten Sitzung ersetzen. Stattdessen werden sie jedoch zusammengeführt und die Verweise auf die Cookies in Index.dat sind infolgedessen veraltet. Weiteres Browsen in neuen Sitzungen kann zum wiederholten Zusammenführen und einem aufgeblähten Cookie-Ordner führen.

Durch Spiegeln des Cookie-Ordners wird dieses Problem gelöst, indem Cookies, jedes Mal wenn der Benutzer sich abmeldet, mit denen der letzten Sitzung überschrieben werden. So bleibt Index.dat aktuell.

Standardmäßig ist diese Einstellung deaktiviert und es werden keine Ordner gespiegelt.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Richtlinie weder hier noch in der INI-Datei konfiguriert ist, werden keine Ordner gespiegelt.

Einstellungen der Richtlinie "Ordnerumleitung"

Der Abschnitt "Ordnerumleitung" enthält Richtlinieneinstellungen für die Angabe, ob häufig in Profilen erscheinende Ordner an einen freigegebenen Speicherort im Netzwerk umgeleitet werden sollen.

Administratorzugriff gewähren

Diese Einstellung ermöglicht einem Administrator für den Zugriff auf den Inhalt von umgeleiteten Ordnern der Benutzer.

Diese Einstellung ist standardmäßig deaktiviert und es haben ausschließlich Benutzer Zugriff auf den Inhalt ihrer umgeleiteten Ordner.

Domänennamen einschließen

Diese Einstellung ermöglicht die Verwendung der Umgebungsvariablen %userdomain% als Teil des für umgeleitete Ordner angegebenen UNC-Pfads.

Standardmäßig ist diese Einstellung deaktiviert und die Umgebungsvariable %userdomain% ist nicht Teil der UNC-Pfad-Angabe für umgeleitete Ordner.

Einstellungen der Richtlinie "AppData(Roaming)"

Der Abschnitt "AppData(Roaming)" enthält Richtlinieneinstellungen für die Angabe, ob der Inhalt des Ordners "AppData(Roaming)" an einen freigegebenen Speicherort im Netzwerk umgeleitet werden soll.

AppData(Roaming)-Pfad

Mit dieser Einstellung geben Sie den Netzwerkspeicherort an, zu dem der Inhalt des Ordners "AppData(Roaming)" umgeleitet werden soll.

Standardmäßig ist diese Einstellung deaktiviert und kein Speicherort wird angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Umleitungseinstellungen für AppData(Roaming)

Mit dieser Einstellung geben Sie an, wie der Inhalt des Ordners "AppData(Roaming)" umgeleitet werden soll.

Standardmäßig erfolgt die Umleitung an einen UNC-Pfad.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Einstellungen der Richtlinie "Kontakte"

Der Abschnitt "Kontakte" enthält Richtlinieneinstellungen für die Angabe, ob der Inhalt des Ordners "Kontakte" an einen freigegebenen Speicherort im Netzwerk umgeleitet werden soll.

'Kontakte'-Pfad

Mit dieser Einstellung geben Sie den Netzwerkspeicherort an, zu dem der Inhalt des Ordners "Kontakte" umgeleitet werden soll.

Standardmäßig ist diese Einstellung deaktiviert und kein Speicherort wird angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Umleitungseinstellungen für 'Kontakte'

Mit dieser Einstellung geben Sie an, wie der Inhalt des Ordners "Kontakte" umgeleitet werden soll.

Standardmäßig erfolgt die Umleitung an einen UNC-Pfad.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Einstellungen der Richtlinie "Desktop"

Der Abschnitt "Desktop" enthält Richtlinieneinstellungen für die Angabe, ob der Inhalt des Ordners "Desktop" an einen freigegebenen Speicherort im Netzwerk umgeleitet werden soll.

'Desktop'-Pfad

Mit dieser Einstellung geben Sie den Netzwerkspeicherort an, zu dem der Inhalt des Ordners "Kontakte" umgeleitet werden soll.

Standardmäßig ist diese Einstellung deaktiviert und kein Speicherort wird angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Umleitungseinstellungen für 'Desktop'

Mit dieser Einstellung geben Sie an, wie der Inhalt des Ordners "Desktop" umgeleitet werden soll.

Standardmäßig erfolgt die Umleitung an einen UNC-Pfad.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Einstellungen der Richtlinie "Dokumente"

Der Abschnitt "Dokumente" enthält Richtlinieneinstellungen für die Angabe, ob der Inhalt des Ordners "Dokumente" an einen freigegebenen Speicherort im Netzwerk umgeleitet werden soll.

'Dokumente'-Pfad

Mit dieser Einstellung geben Sie den Netzwerkspeicherort an, zu dem die Dateien im Ordner "Dokumente" umgeleitet werden sollen.

Standardmäßig ist diese Einstellung deaktiviert und kein Speicherort wird angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Umleitungseinstellungen für 'Dokumente'

Mit dieser Einstellung geben Sie an, wie der Inhalt des Ordners "Dokumente" umgeleitet werden soll.

Standardmäßig erfolgt die Umleitung an einen UNC-Pfad.

Wählen Sie eine der folgenden Optionen, um zu steuern, wie der Inhalt des Ordners "Dokumente" umgeleitet werden soll:

- Zum folgenden UNC-Pfad umleiten: leitet den Inhalt zu dem in der Richtlinieneinstellung 'Dokumente'-Pfad angegebenen UNC-Pfad.
- Zum Basisverzeichnis des Benutzers umleiten: leitet den Inhalt zu dem Basisverzeichnis des Benutzers. Dieses ist normalerweise mit dem Attribut #homeDirectory# für einen Benutzer in Active Directory konfiguriert.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Einstellungen der Richtlinie "Downloads"

Der Abschnitt "Downloads" enthält Richtlinieneinstellungen für die Angabe, ob der Inhalt des Ordners "Downloads" an einen freigegebenen Speicherort im Netzwerk umgeleitet werden soll.

'Downloads'-Pfad

Mit dieser Einstellung geben Sie den Netzwerkspeicherort an, zu dem die Dateien im Ordner "Downloads" umgeleitet werden.

Standardmäßig ist diese Einstellung deaktiviert und kein Speicherort wird angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Umleitungseinstellungen für 'Downloads'

Mit dieser Einstellung geben Sie an, wie der Inhalt des Ordners "Downloads" umgeleitet werden soll.

Standardmäßig erfolgt die Umleitung an einen UNC-Pfad.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Einstellungen der Richtlinie "Favoriten"

Der Abschnitt "Favoriten" enthält Richtlinieneinstellungen für die Angabe, ob der Inhalt des Ordners "Favoriten" an einen freigegebenen Speicherort im Netzwerk umgeleitet werden soll.

'Favoriten'-Pfad

Mit dieser Einstellung geben Sie die Netzwerkfreigabe an, zu der der Inhalt des Ordners "Favoriten" umgeleitet werden soll.

Standardmäßig ist diese Einstellung deaktiviert und kein Speicherort wird angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Umleitungseinstellungen für 'Favoriten'

Mit dieser Einstellung geben Sie an, wie der Inhalt des Ordners "Favoriten" umgeleitet werden soll.

Standardmäßig erfolgt die Umleitung an einen UNC-Pfad.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Einstellungen der Richtlinie "Links"

Der Abschnitt "Links" enthält Richtlinieneinstellungen für die Angabe, ob der Inhalt des Ordners "Links" an einen freigegebenen Speicherort im Netzwerk umgeleitet werden soll.

'Links'-Pfad

Mit dieser Einstellung geben Sie die Netzwerkfreigabe an, zu der der Inhalt des Ordners "Links" umgeleitet werden soll.

Standardmäßig ist diese Einstellung deaktiviert und kein Speicherort wird angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Umleitungseinstellungen für 'Links'

Mit dieser Einstellung geben Sie an, wie der Inhalt des Ordners "Links" umgeleitet werden soll.

Standardmäßig erfolgt die Umleitung an einen UNC-Pfad.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Einstellungen der Richtlinie "Musik"

Der Abschnitt "Musik" enthält Richtlinieneinstellungen für die Angabe, ob der Inhalt des Ordners "Musik" an einen freigegebenen Speicherort im Netzwerk umgeleitet werden soll.

'Musik'-Pfad

Mit dieser Einstellung geben Sie die Netzwerkfreigabe an, zu der der Inhalt des Ordners "Musik" umgeleitet werden soll.

Standardmäßig ist diese Einstellung deaktiviert und kein Speicherort wird angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Umleitungseinstellungen für 'Musik'

Mit dieser Einstellung geben Sie an, wie der Inhalt des Ordners "Musik" umgeleitet werden soll.

Standardmäßig erfolgt die Umleitung an einen UNC-Pfad.

Wählen Sie eine der folgenden Optionen, um zu steuern, wie der Inhalt des Ordners "Musik" umgeleitet werden soll:

- Zum folgenden UNC-Pfad umleiten: Leitet den Inhalt zu dem in der Richtlinieneinstellung 'Musik'-Pfad angegebenen UNC-Pfad.
- Relativ zum Ordner 'Dokumente' umleiten: Leitet den Inhalt in einen Ordner relativ zu dem Ordner "Dokumente" um.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Einstellungen der Richtlinie "Bilder"

Der Abschnitt "Bilder" enthält Richtlinieneinstellungen für die Angabe, ob der Inhalt des Ordners "Bilder" an einen freigegebenen Speicherort im Netzwerk umgeleitet werden soll.

'Bilder'-Pfad

Mit dieser Einstellung geben Sie die Netzwerkfreigabe an, zu der der Inhalt des Ordners "Bilder" umgeleitet werden soll.

Standardmäßig ist diese Einstellung deaktiviert und kein Speicherort wird angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Umleitungseinstellungen für 'Bilder'

Mit dieser Einstellung geben Sie an, wie der Inhalt des Ordners "Bilder" umgeleitet werden soll.

Standardmäßig erfolgt die Umleitung an einen UNC-Pfad.

Wählen Sie eine der folgenden Optionen, um zu steuern, wie der Inhalt des Ordners "Bilder" umgeleitet werden soll:

- Zum folgenden UNC-Pfad umleiten: Leitet den Inhalt zu dem in der Richtlinieneinstellung 'Bilder'-Pfad angegebenen UNC-Pfad.
- Relativ zum Ordner 'Dokumente' umleiten: Leitet den Inhalt in einen Ordner relativ zu dem Ordner "Dokumente" um.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Einstellungen der Richtlinie "Gespeicherte Spiele"

Der Abschnitt "Gespeicherte Spiele" enthält Richtlinieneinstellungen für die Angabe, ob der Inhalt des Ordners "Gespeicherte Spiele" an einen freigegebenen Speicherort im Netzwerk umgeleitet werden soll.

Umleitungseinstellungen für 'Gespeicherte Spiele'

Mit dieser Einstellung geben Sie an, wie der Inhalt des Ordners "Gespeicherte Spiele" umgeleitet werden soll.

Standardmäßig erfolgt die Umleitung an einen UNC-Pfad.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

'Gespeicherte Spiele'-Pfad

Mit dieser Einstellung geben Sie die Netzwerkfreigabe an, zu der der Inhalt des Ordners "Gespeicherte Spiele" umgeleitet werden soll.

Standardmäßig ist diese Einstellung deaktiviert und kein Speicherort wird angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Einstellungen der Richtlinie "Suchen"

Der Abschnitt "Suchen" enthält Richtlinieneinstellungen für die Angabe, ob der Inhalt des Ordners "Suchen" an einen freigegebenen Speicherort im Netzwerk umgeleitet werden soll.

Umleitungseinstellungen für 'Suchen'

Mit dieser Einstellung geben Sie an, wie der Inhalt des Ordners "Suchen" umgeleitet werden soll.

Standardmäßig erfolgt die Umleitung an einen UNC-Pfad.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

'Suchen'-Pfad

Mit dieser Einstellung geben Sie die Netzwerkfreigabe an, zu der der Inhalt des Ordners "Suchen" umgeleitet werden soll.

Standardmäßig ist diese Einstellung deaktiviert und kein Speicherort wird angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Einstellungen der Richtlinie "Startmenü"

Der Abschnitt "Startmenü" enthält Richtlinieneinstellungen für die Angabe, ob der Inhalt des Ordners "Startmenü" an einen freigegebenen Speicherort im Netzwerk umgeleitet werden soll.

Umleitungseinstellungen für 'Startmenü'

Mit dieser Einstellung geben Sie an, wie der Inhalt des Ordners "Startmenü" umgeleitet werden soll.

Standardmäßig erfolgt die Umleitung an einen UNC-Pfad.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Startmenü-Pfad

Mit dieser Einstellung geben Sie die Netzwerkfreigabe an, zu der der Inhalt des Ordners "Startmenü" umgeleitet werden soll.

Standardmäßig ist diese Einstellung deaktiviert und kein Speicherort wird angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Einstellungen der Richtlinie "Videos"

Der Abschnitt "Videos" enthält Richtlinieneinstellungen für die Angabe, ob der Inhalt des Ordners "Videos" an einen freigegebenen Speicherort im Netzwerk umgeleitet werden soll.

Umleitungseinstellungen für 'Videos'

Mit dieser Einstellung geben Sie an, wie der Inhalt des Ordners "Videos" umgeleitet werden soll.

Standardmäßig erfolgt die Umleitung an einen UNC-Pfad.

Wählen Sie eine der folgenden Optionen, um zu steuern, wie der Inhalt des Ordners "Videos" umgeleitet werden soll:

- Zum folgenden UNC-Pfad umleiten: Leitet den Inhalt zu dem in der Richtlinieneinstellung 'Videos'-Pfad angegebenen UNC-Pfad.
- Relativ zum Ordner 'Dokumente' umleiten: Leitet den Inhalt in einen Ordner relativ zu dem Ordner "Dokumente" um.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

'Videos'-Pfad

Mit dieser Einstellung geben Sie die Netzwerkfreigabe an, zu der der Inhalt des Ordners "Videos" umgeleitet werden soll.

Standardmäßig ist diese Einstellung deaktiviert und kein Speicherort wird angegeben.

Wenn diese Einstellung hier nicht konfiguriert ist, erfolgt keine Umleitung durch die Profilverwaltung.

Einstellungen der Richtlinie "Protokollierung"

Der Abschnitt "Protokollierung" enthält Richtlinieneinstellungen zum Konfigurieren der Protokollierung der Profilverwaltung.

Active Directory-Aktionen

Mit dieser Einstellung aktivieren oder deaktivieren Sie die ausführliche Protokollierung der in Active Directory ausgeführten Aktionen.

Diese Einstellung ist standardmäßig deaktiviert. Wenn Sie diese Einstellung aktivieren, stellen Sie sicher, dass die Einstellung Protokollierung aktivieren auch aktiviert ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Fehler und allgemeine Informationen protokolliert.

Allgemeine Informationen

Mit dieser Einstellung aktivieren oder deaktivieren Sie die ausführliche Protokollierung allgemeiner Informationen.

Diese Einstellung ist standardmäßig deaktiviert. Wenn Sie diese Einstellung aktivieren, stellen Sie sicher, dass die Einstellung Protokollierung aktivieren auch aktiviert ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Fehler und allgemeine Informationen protokolliert.

Allgemeine Warnungen

Mit dieser Einstellung aktivieren oder deaktivieren Sie die ausführliche Protokollierung allgemeiner Warnungen.

Diese Einstellung ist standardmäßig deaktiviert. Wenn Sie diese Einstellung aktivieren, stellen Sie sicher, dass die Einstellung Protokollierung aktivieren auch aktiviert ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Fehler und allgemeine Informationen protokolliert.

Protokollierung aktivieren

Mit dieser Einstellung aktivieren oder deaktivieren Sie die Protokollierung der Profilverwaltung im Debugmodus (ausführliche Protokollierung). Im Debugmodus werden umfangreiche Statusinformationen in den Protokolldateien unter "%SystemRoot%\System32\Logfiles\UserProfileManager" aufgezeichnet.

Standardmäßig ist diese Einstellung deaktiviert und es werden nur Fehler protokolliert. Citrix empfiehlt, dass Sie diese Einstellung nur aktivieren, wenn Sie eine Problembehandlung für die Profilverwaltung durchführen.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden nur Fehler protokolliert.

Dateisystemaktionen

Mit dieser Einstellung aktivieren oder deaktivieren Sie die ausführliche Protokollierung der im Dateisystem ausgeführten Aktionen.

Diese Einstellung ist standardmäßig deaktiviert. Wenn Sie diese Einstellung aktivieren, stellen Sie sicher, dass die Einstellung Protokollierung aktivieren auch aktiviert ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Fehler und allgemeine Informationen protokolliert.

Dateisystembenachrichtigungen

Mit dieser Einstellung aktivieren oder deaktivieren Sie die ausführliche Protokollierung von Dateisystembenachrichtigungen.

Diese Einstellung ist standardmäßig deaktiviert. Wenn Sie diese Einstellung aktivieren, stellen Sie sicher, dass die Einstellung Protokollierung aktivieren auch aktiviert ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Fehler und allgemeine Informationen protokolliert.

Abmeldung

Mit dieser Einstellung aktivieren oder deaktivieren Sie die ausführliche Protokollierung von Benutzerabmeldungen.

Diese Einstellung ist standardmäßig deaktiviert. Wenn Sie diese Einstellung aktivieren, stellen Sie sicher, dass die Einstellung Protokollierung aktivieren auch aktiviert ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Fehler und allgemeine Informationen protokolliert.

Anmeldung

Mit dieser Einstellung aktivieren oder deaktivieren Sie die ausführliche Protokollierung von Benutzeranmeldungen.

Diese Einstellung ist standardmäßig deaktiviert. Wenn Sie diese Einstellung aktivieren, stellen Sie sicher, dass die Einstellung Protokollierung aktivieren auch aktiviert ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Fehler und allgemeine Informationen protokolliert.

Max. Größe der Protokolldatei

Mit dieser Einstellung geben Sie die maximal zulässige Größe für die Protokolldatei der Profilverwaltung in Bytes an.

Der Standardwert dieser Einstellung ist 1048576 Bytes (1 MB). Citrix empfiehlt, dass die Größe dieser Datei auf 5 MB oder mehr erhöht wird, sofern Sie ausreichend Speicherplatz auf dem Datenträger haben. Wenn die Protokolldatei die maximale Größe überschreitet, wird eine vorhandene Sicherungskopie der Datei (.bak) gelöscht, die Protokolldatei erhält die Erweiterung .bak und eine neue Protokolldatei wird erstellt. Die Protokolldatei wird unter "%SystemRoot%\System32\Logfiles\UserProfileManager" erstellt.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, wird der Standardwert verwendet.

Pfad zur Protokolldatei

Mit dieser Einstellung geben Sie einen alternativen Pfad an, der zum Speichern der Protokolldatei der Profilverwaltung verwendet wird.

Standardmäßig ist diese Einstellung deaktiviert und Protokolldateien werden am Standardspeicherort %SystemRoot%\System32\Logfiles\UserProfileManager gespeichert.

Der Pfad kann zu einem lokalen Laufwerk oder einem Remotelaufwerk im Netzwerk (UNC-Pfad) führen. Remotepfade können in großen, verteilten Umgebungen von Nutzen sein, können aber zu erheblichem Netzwerkverkehr führen, was für Protokolldateien unangebracht ist. Geben Sie für bereitgestellte virtuelle Maschinen mit einer beständigen Festplatte einen lokalen Pfad zu diesem Laufwerk an. Hierdurch wird sichergestellt, dass Protokolldateien beim Neustart der Maschine beibehalten werden. Geben Sie für virtuelle Maschinen ohne eine beständige Festplatte einen UNC-Pfad an. So werden Protokolldateien beibehalten. Das Systemkonto für die Maschinen muss aber Schreibzugriff auf die UNC-Freigabe haben. Verwenden Sie für Laptops, die vom Feature für Offlineprofile verwaltet werden, einen lokalen Pfad.

Wenn für Protokolldateien ein UNC-Pfad verwendet wird, empfiehlt Citrix, entsprechende Zugriffssteuerungslisten auf den Ordner mit den Protokolldateien anzuwenden, um sicherzustellen, dass nur autorisierte Benutzer- oder Computerkonten auf die gespeicherten Dateien zugreifen können.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, wird der Standardspeicherort "%SystemRoot%\System32\Logfiles\UserProfileManager" verwendet.

Persönliche Benutzerinformationen

Mit dieser Einstellung aktivieren oder deaktivieren Sie die ausführliche Protokollierung persönlicher Benutzerinformationen.

Diese Einstellung ist standardmäßig deaktiviert. Wenn Sie diese Einstellung aktivieren, stellen Sie sicher, dass die Einstellung Protokollierung aktivieren auch aktiviert ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Fehler und allgemeine Informationen protokolliert.

Richtlinienwerte bei Anmeldung und Abmeldung

Mit dieser Einstellung aktivieren oder deaktivieren Sie ausführliche Protokollierung der Richtlinienwerte beim An- und Abmelden von Benutzern.

Diese Einstellung ist standardmäßig deaktiviert. Wenn Sie diese Einstellung aktivieren, stellen Sie sicher, dass die Einstellung Protokollierung aktivieren auch aktiviert ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Fehler und allgemeine Informationen protokolliert.

Registrierungsaktionen

Mit dieser Einstellung aktivieren oder deaktivieren Sie die ausführliche Protokollierung der in der Registrierung ausgeführten Aktionen.

Diese Einstellung ist standardmäßig deaktiviert. Wenn Sie diese Einstellung aktivieren, stellen Sie sicher, dass die Einstellung Protokollierung aktivieren auch aktiviert ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Fehler und allgemeine Informationen protokolliert.

Registrierungsunterschiede bei der Abmeldung

Mit dieser Einstellung aktivieren oder deaktivieren Sie ausführliche Protokollierung aller Registrierungsunterschiede bei der Abmeldung von Benutzern.

Diese Einstellung ist standardmäßig deaktiviert. Wenn Sie diese Einstellung aktivieren, stellen Sie sicher, dass die Einstellung Protokollierung aktivieren auch aktiviert ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden Fehler und allgemeine Informationen protokolliert.

Einstellungen der Richtlinie "Profilverarbeitung"

Der Abschnitt "Profilverarbeitung" enthält Richtlinieneinstellungen zum Konfigurieren der Verarbeitung von Benutzerprofilen durch die Profilverwaltung.

Verzögerung vor dem Löschen von zwischengespeicherten Profilen

Mit dieser Einstellung geben Sie optional eine Verlängerung für die Verzögerung (in Minuten) ein, nach der die Profilverwaltung lokal zwischengespeicherte Profile bei der Abmeldung löscht.

Bei einem Wert von 0 werden die Profile am Ende der Abmeldung sofort gelöscht. Die Profilverwaltung prüft jede Minute auf Abmeldungen, sodass ein Wert von 60 sicherstellt, dass Profile zwischen einer und zwei Minuten (je nachdem, wann die letzte Überprüfung stattgefunden hat) nach dem Abmelden gelöscht werden. Das Erweitern der Verzögerung ist nützlich, wenn Sie wissen, dass ein Prozess Dateien oder die Registrierungsstruktur während der Abmeldung geöffnet hält. Bei großen Profilen kann dies auch den Abmeldungsprozess beschleunigen.

Die Standardeinstellung ist 0, lokal zwischengespeicherte Profile werden von der Profilverwaltung sofort gelöscht. Wenn Sie diese Einstellung aktivieren, müssen Sie sicherstellen, dass die Einstellung Lokal zwischengespeicherte Profile nach Abmeldung löschen auch aktiviert ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden die Profile sofort gelöscht.

Lokal zwischengespeicherte Profile nach Abmeldung löschen

Mit dieser Einstellung geben Sie an, ob lokal zwischengespeicherte Profile gelöscht werden, nachdem Benutzer sich abmelden.

Wenn diese Einstellung aktiviert ist, wird der lokale Profilcache der Benutzer nach der Abmeldung gelöscht. Citrix empfiehlt, dass Sie diese Einstellung für Terminalserver aktivieren.

Standardmäßig ist diese Einstellung deaktiviert und der lokale Profilcache von Benutzern wird nach der Abmeldung beibehalten.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden zwischengespeicherte Profile nicht gelöscht.

Behandlung von Konflikten lokaler Profile

Mit dieser Einstellung wird konfiguriert, wie die Profilverwaltung verfährt, wenn ein Benutzerprofil sowohl im Benutzerspeicher als auch als lokales Windows-Benutzerprofil (kein Citrix Benutzerprofil) vorhanden ist.

Standardmäßig verwendet die Profilverwaltung lokale Windows-Profile, ohne diese jedoch zu ändern.

Zum Steuern, wie die Profilverwaltung verfahren soll, wählen Sie eine der folgenden Optionen:

- Lokales Profil verwenden: Die Profilverwaltung verwendet lokale Windows-Profile, ohne diese jedoch zu ändern.
- Lokales Profil löschen: Die Profilverwaltung löscht das lokale Windows-Benutzerprofil und importiert dann das Citrix Benutzerprofil aus dem Benutzerspeicher.
- Lokales Profil umbenennen: Die Profilverwaltung benennt das lokale Windows-Benutzerprofil um (als Sicherungskopie) und importiert dann das Citrix Benutzerprofil aus dem Benutzerspeicher.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden vorhandene lokale Profile verwendet.

Migration vorhandener Profile

Mit dieser Einstellung geben Sie den Typ des Profils an, das bei der Anmeldung eines Benutzers in den Benutzerspeicher migriert wird, wenn der Speicher kein aktuelles Profil für den Benutzer enthält.

Die Profilverwaltung kann vorhandene Profile während der Anmeldung spontan migrieren, wenn der Benutzer kein Profil im Benutzerspeicher hat. Anschließend verwendet die Profilverwaltung das Profil im Benutzerspeicher in der aktuellen Sitzung und in allen künftigen Sitzungen, die mit dem Pfad zu demselben Benutzerspeicher konfiguriert sind.

Standardmäßig werden lokale Profile und Roamingprofile während der Anmeldung in den Benutzerspeicher migriert.

Um anzugeben welche Profiltypen bei der Anmeldung in den Benutzerspeicher migriert werden sollen, wählen Sie eine der folgenden Optionen:

- Lokal und Roaming
- Lokal
- Roaming
- Keine (deaktiviert)

Wenn Sie Keine auswählen, wird der vorhandene Windows-Mechanismus für die Erstellung neuer Profile verwendet, genau wie in einer Umgebung, in der die Profilverwaltung nicht installiert ist.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden vorhandene lokale und servergespeicherte Profile migriert.

Pfad zum Vorlagenprofil

Mit dieser Einstellung geben Sie den Pfad zu dem Profil an, das die Profilverwaltung als Vorlage zum Erstellen neuer Benutzerprofile verwenden soll.

Dies muss der vollständige Pfad zu dem Ordner sein, der die Registrierungsdatei NTUSER.DAT und sämtliche anderen für das Vorlagenprofil erforderlichen Dateien und Ordner enthält.

Hinweis: Geben Sie das Element NTUSER.DAT nicht mit dem Pfad ein. Geben Sie für die Datei `\\Server\Profile\Vorlage\ntuser.dat` den Speicherort als `\\Server\Profile\Vorlage` an.

Verwenden Sie einen absoluten Pfad (entweder einen UNC-Pfad oder einen Pfad auf dem lokalen Computer). Sie können einen lokalen Pfad verwenden, um z. B. ein Vorlagenprofil auf einem Citrix Provisioning Services-Image dauerhaft anzugeben. Relative Pfade werden nicht unterstützt.

Hinweis: Beachten Sie, dass diese Richtlinie nicht die Erweiterung von Active Directory-Attributen, Systemumgebungsvariablen oder der Variablen `%USERNAME%` und `%USERDOMAIN%` unterstützt.

Standardmäßig ist diese Einstellung deaktiviert und neue Benutzerprofile werden auf der Basis des Standardbenutzerprofils auf dem Gerät, auf dem sich der Benutzer als erstes anmeldet, erstellt.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden keine Vorlagen verwendet.

Vorlagenprofil überschreibt lokales Profil

Diese Einstellung ermöglicht eine Überschreibung des lokalen Profils durch das Vorlagenprofil bei der Erstellung neuer Benutzerprofile.

Wenn ein Benutzer kein Citrix Benutzerprofil hat, aber ein lokales Windows-Benutzerprofil vorhanden ist, wird standardmäßig das lokale Profil verwendet (und in den Benutzerspeicher migriert, wenn diese Option nicht deaktiviert ist). Durch Aktivieren dieser Einstellung kann das Vorlagenprofil das lokale Profil bei der Erstellung neuer Benutzerprofile

überschreiben.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden keine Vorlagen verwendet.

Vorlagenprofil überschreibt Roamingprofil

Diese Einstellung ermöglicht eine Überschreibung eines Roamingprofils durch das Vorlagenprofil bei der Erstellung neuer Benutzerprofile.

Wenn ein Benutzer kein Citrix Benutzerprofil hat aber ein lokales Windows-Benutzerprofil vorhanden ist, wird standardmäßig das Roamingprofil verwendet (und in den Benutzerspeicher migriert, wenn diese Option nicht deaktiviert ist). Durch Aktivieren dieser Einstellung kann das Vorlagenprofil das Roamingprofil bei der Erstellung neuer Benutzerprofile überschreiben.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden keine Vorlagen verwendet.

Als verbindliches Citrix Profil für alle Anmeldungen verwendete Vorlagenprofil

Bei Auswahl dieser Einstellung verwendet die Profilverwaltung das Vorlagenprofil als Standardprofil bei der Erstellung aller neuen Benutzerprofile.

Standardmäßig ist diese Einstellung deaktiviert und neue Benutzerprofile werden auf der Basis des Standardbenutzerprofils auf dem Gerät, auf dem sich der Benutzer als erstes anmeldet, erstellt.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden keine Vorlagen verwendet.

Einstellungen der Richtlinie "Registrierung"

Der Abschnitt "Registrierung" enthält Richtlinieneinstellungen, mit denen Sie festlegen können, welche Registrierungsschlüssel bei der Verarbeitung der Profilverwaltung berücksichtigt und welche ausgeschlossen werden sollen.

Ausschlussliste

Mit dieser Einstellung geben Sie die Liste der Registrierungsschlüssel in der HKCU-Struktur an, die nicht von der Profilverwaltung verarbeitet werden sollen, wenn sich ein Benutzer abmeldet.

Wenn diese Option aktiviert ist, werden die in dieser Liste aufgeführten Schlüssel von der Verarbeitung ausgeschlossen, wenn sich ein Benutzer abmeldet.

Standardmäßig ist diese Einstellung deaktiviert, und alle Registrierungsschlüssel in der HKCU-Struktur werden verarbeitet.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden keine Registrierungsschlüssel von der Verarbeitung ausgeschlossen.

Aufnahmeliste

Mit dieser Einstellung geben Sie die Liste der Registrierungsschlüssel in der HKCU-Struktur an, die von der Profilverwaltung

verarbeitet werden sollen, wenn sich ein Benutzer abmeldet.

Wenn diese Option aktiviert ist, werden die in dieser Liste aufgeführten Schlüssel in die Verarbeitung eingeschlossen, wenn sich ein Benutzer abmeldet.

Standardmäßig ist diese Einstellung deaktiviert, und alle Registrierungsschlüssel in der HKCU-Struktur werden verarbeitet.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet.

Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, wird die gesamte HKCU-Struktur verarbeitet.

Einstellungen der Richtlinie "Gestreamte Benutzerprofile"

Der Abschnitt "Gestreamte Benutzerprofile" enthält Richtlinieneinstellungen zum Konfigurieren der Verarbeitung von gestreamten Benutzerprofilen durch die Profilverwaltung.

Immer zwischenspeichern

Mit dieser Einstellung geben Sie an, ob die Profilverwaltung gestreamte Dateien so bald wie möglich zwischenspeichern soll, wenn sich ein Benutzer anmeldet. Durch Zwischenspeichern von Dateien nachdem sich ein Benutzer anmeldet werden Netzwerkbandbreite gespart und die Benutzererfahrung optimiert.

Verwenden Sie diese Einstellung in Verbindung mit der Einstellung Profilstreaming. Standardmäßig ist diese Einstellung deaktiviert und gestreamte Dateien werden nicht so schnell wie möglich zwischengespeichert, wenn sich ein Benutzer anmeldet.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, ist sie deaktiviert.

Immer Cachegröße

Mit dieser Einstellung geben Sie eine Untergrenze (in Megabytes) für die Größe der Dateien an, die gestreamt werden. Die Profilverwaltung speichert Dateien dieser Größe bzw. größere Dateien so bald wie möglich zwischen, wenn ein Benutzer sich anmeldet.

Die Standardeinstellung ist 0 (null) und die Funktion zum Zwischenspeichern des gesamten Profils wird verwendet. Wenn das Feature zum Zwischenspeichern des gesamten Profils aktiviert ist, ruft die Profilverwaltung den gesamten Inhalt des Profils im Benutzerspeicher als Hintergrundaufgabe ab, nachdem sich ein Benutzer anmeldet.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, ist sie deaktiviert.

Profilstreaming

Diese Einstellung aktiviert oder deaktiviert das Feature für gestreamte Citrix Benutzerprofile. Wenn diese Option aktiviert ist, werden Dateien und Ordner in einem Profil nur dann aus dem Benutzerspeicher auf den lokalen Computer abgerufen, wenn auf sie von Benutzern nach der Anmeldung zugegriffen wird. Registrierungseinträge und alle Dateien im Bereich für ausstehende Dateien werden sofort abgerufen.

Standardmäßig ist das Profilstreaming deaktiviert.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder

hier noch in der INI-Datei konfiguriert ist, ist sie deaktiviert.

Gruppen mit gestreamten Benutzerprofilen

Mit dieser Einstellung geben Sie die Benutzerprofile in einer Organisationseinheit gestreamt werden, basierend auf Windows Benutzergruppen.

Wenn diese Option aktiviert ist, werden nur die Benutzerprofile in den angegebenen Benutzergruppen gestreamt. Alle anderen Benutzerprofile werden normal verarbeitet. Standardmäßig ist diese Einstellung deaktiviert und alle Dateien in Benutzerprofilen werden normal verarbeitet.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, werden alle Benutzergruppen verarbeitet.

Timeout für gesperrte Dateien im ausstehenden Bereich

Mit dieser Einstellung geben Sie einen Zeitraum (in Tagen) an, nach dem Dateien der Benutzer aus dem Bereich für ausstehende Dateien in den Benutzerspeicher zurückgeschrieben werden, wenn ein Server nicht mehr reagiert und der Benutzerspeicher gesperrt bleibt. Dies verhindert ein Aufblähen des ausstehenden Bereichs und stellt sicher, dass der Benutzerstore immer die aktuellen Dateien enthält.

Die Standardeinstellung ist 1 Tag.

Wenn diese Einstellung hier nicht konfiguriert ist, wird der Wert in der INI-Datei verwendet. Wenn diese Einstellung weder hier noch in der INI-Datei konfiguriert ist, wird der Standardwert verwendet.

Einstellungen der Richtlinie "Receiver"

Sep 29, 2015

Der Abschnitt "Receiver" enthält Richtlinieneinstellungen zum Konfigurieren einer Liste von Storefront-Adressen, die für auf dem virtuellen Desktop ausgeführte Windows-Betriebssysteme per Push an Receiver für Windows übertragen werden sollen.

StoreFront-Kontenliste

Mit dieser Einstellung geben Sie eine Liste der StoreFront-Stores an, die Administratoren für die Übertragung per Push an Receiver für Windows auswählen können. Beim Erstellen einer Bereitstellungsgruppe können Administratoren auswählen, welche Stores für Receiver für Windows auf virtuellen Desktops, der in dieser Gruppe ausgeführt werden, per Push übertragen werden sollen.

In der Standardeinstellung sind keine Stores angegeben.

Geben Sie für jeden Store die folgenden Informationen in Form eines durch Semikolons getrennten Eintrags an:

- Storename: der Name des Stores wie er Benutzern angezeigt wird.
- Store-URL: die URL des Stores.
- Storeaktivierungszustand: gibt an, ob der Store für Benutzer verfügbar ist. Dies ist entweder ein oder aus.
- Store-Beschreibung: die Beschreibung des Stores, die Benutzern angezeigt wird.

Beispiel:

Verkaufs-Store;https://sales.mycompany.com/Citrix/Store/discovery;On;Store für Verkaufsmitarbeiter

Einstellungen der Richtlinie "Virtual Delivery Agent"

Sep 29, 2015

Der Abschnitt "Virtual Delivery Agent" enthält Richtlinieneinstellungen, mit denen Sie die Kommunikation zwischen VDA und Controllern einer Site steuern können.

Wichtig: Der VDA benötigt die in diesen Einstellungen enthaltenen Informationen für die Registrierung bei einem Controller, wenn das Feature für automatische Controllerupdates nicht verwendet wird. Da die Informationen für die Registrierung erforderlich sind, müssen Sie sie mit dem Active Directory-Gruppenrichtlinien-Editor konfigurieren, sofern Sie sie nicht bei der VDA-Installation angeben.

- IPv6-Netzwerkmaske für Controllerregistrierung
- Controllerregistrierungsport
- Controller-SIDs
- Controller
- Nur IPv6-Controllerregistrierung verwenden
- Site-GUID

IPv6-Netzwerkmaske für Controllerregistrierung

Mit dieser Richtlinieneinstellung kann der VDA auf ein bevorzugtes Subnetz (anstelle einer globalen IP, sofern registriert) limitiert werden. Mit dieser Einstellung geben Sie die IPv6-Adresse und das Netzwerk an, in dem der VDA registriert wird. Der VDA wird nur an der ersten Adresse registriert, die mit der angegebenen Netzmaske übereinstimmt. Diese Einstellung ist nur gültig, wenn die Richtlinieneinstellung Nur IPv6-Controllerregistrierung verwenden aktiviert ist.

Diese Einstellung ist standardmäßig leer.

Controllerregistrierungsport

Verwenden Sie diese Einstellung nur, wenn die Einstellung Automatische Controllerupdates aktivieren deaktiviert ist.

Mit dieser Einstellung geben Sie die TCP/IP-Portnummer an, die der VDA für die Registrierung bei einem Controller verwendet, wenn die registrierungsbasierte Registrierung verwendet wird.

Die Standardeinstellung der Portnummer ist "80".

Controller-SIDs

Verwenden Sie diese Einstellung nur, wenn die Einstellung Automatische Controllerupdates aktivieren deaktiviert ist.

Mit dieser Einstellung geben Sie eine durch Leerzeichen getrennte Liste von Controller-SIDs an, die der VDA für die Registrierung bei einem Controller verwendet, wenn die registrierungsbasierte Registrierung verwendet wird. Dies ist eine optionale Einstellung, die im Zusammenhang mit der Einstellung "Controller" verwendet werden kann, um die für die Registrierung verwendete Liste von Controllern zu beschränken.

Diese Einstellung ist standardmäßig leer.

Controller

Verwenden Sie diese Einstellung nur, wenn die Einstellung Automatische Controllerupdates aktivieren deaktiviert ist.

Mit dieser Einstellung geben Sie eine durch Leerzeichen getrennte Liste von vollständig qualifizierten Domännennamen

(FQDN) für Controller an, die der VDA für die Registrierung bei einem Controller verwendet, wenn die registrierungsbasierte Registrierung verwendet wird. Dies ist eine optionale Einstellung, die im Zusammenhang mit der Einstellung "Controller-SIDs" verwendet werden kann.

Diese Einstellung ist standardmäßig leer.

Automatische Controllerupdates aktivieren

Mit dieser Einstellung ist eine automatische Registrierung des VDAs bei einem Controller nach der Installation möglich.

Nach der Registrierung wird von dem Controller, bei dem der VDA registriert ist, eine Liste der aktuellen Controller-FQDNs und -SIDs an den VDA gesendet. Diese Liste wird in den persistenten Speicher des VDAs geschrieben. Jeder Controller prüft die Datenbank alle 90 Minuten auf Controllerinformationen. Wurde seit der letzten Prüfung ein Controller hinzugefügt oder entfernt oder ist eine Richtlinienänderung erfolgt, sendet der Controller eine aktualisierte Liste an die bei ihm registrierten VDAs. Der VDA nimmt alle Verbindungen von allen Controllern in der aktuellen Liste an.

Standardmäßig ist diese Einstellung aktiviert.

Nur IPv6-Controllerregistrierung verwenden

Diese Einstellung steuert die Form der Adresse, die vom VDA für die Registrierung beim Controller verwendet wird:

- Bei aktivierter Einstellung wird der VDA mit der IPv6-Adresse der Maschine für die Kommunikation beim Controller registriert. Wenn der VDA mit dem Controller kommuniziert, wird eine Adresse verwendet, deren Auswahl folgender Reihenfolge unterliegt: globale IP-Adresse, ULA-Adresse, Link-Local-Adresse (wenn keine anderen IPv6-Adressen verfügbar sind).
- Bei deaktivierter Einstellung wird der VDA mit der IPv4-Adresse der Maschine für die Kommunikation beim Controller registriert.

Diese Einstellung ist standardmäßig deaktiviert.

Site-GUID

Verwenden Sie diese Einstellung nur, wenn die Einstellung Automatische Controllerupdates aktivieren deaktiviert ist.

Diese Einstellung gibt den Globally Unique Identifier (GUID) der Site an, den der VDA für die Registrierung bei einem Controller verwendet, wenn die Active Directory-basierte Registrierung verwendet wird.

Diese Einstellung ist standardmäßig leer.

Einstellungen der Richtlinie "HDX 3D Pro"

Der Bereich "HDX 3D Pro" enthält Richtlinieneinstellungen, mit denen Sie das Tool zum Konfigurieren der Bildqualität für die Benutzer aktivieren und konfigurieren können. Mit dem Tool können Benutzer die Verwendung der verfügbaren Bandbreite durch Anpassen des Verhältnisses zwischen Bildqualität und Reaktionszeit in Echtzeit optimieren.

Verlustfrei aktivieren

Mit dieser Einstellung wird angegeben, ob Benutzer verlustfreie Komprimierung mit dem Tool zum Konfigurieren der Bildqualität aktivieren oder deaktivieren können. In der Standardeinstellung wird den Benutzern die Möglichkeit zum Aktivieren der verlustfreien Komprimierung nicht eingeräumt.

Aktiviert ein Benutzer die verlustfreie Komprimierung, wird die Bildqualität automatisch auf den höchsten Wert eingestellt,

der im Bildkonfigurationstool verfügbar ist. Standardmäßig kann je nach Leistungsfähigkeit des Benutzergeräts und des Hostcomputers entweder die GPU-basierte oder die CPU-basierte Komprimierung verwendet werden.

HDX3DPro-Qualitätseinstellungen

Mit dieser Einstellung geben Sie den Mindest- und den Höchstwert an, mit denen der Bildqualitätsanpassungsbereich, der den Benutzern im Tool zum Konfigurieren der Bildqualität zur Verfügung steht, festgelegt wird.

Geben Sie für die Bildqualität Werte zwischen 0 und 100 an. Der Höchstwert muss größer oder gleich dem Mindestwert sein.

Konfigurieren von COM-Port- und LPT-Portumleitungseinstellungen in der Registrierung

Sep 29, 2015

Citrix beabsichtigt, die Unterstützung für COM und LPT in einem zukünftigen großen Release einzustellen. Aus diesem Grund wurden die Richtlinieneinstellungen für COM-Port- und LPT-Portumleitung aus Studio in die Registrierung verschoben und befinden sich jetzt unter den folgenden Registrierungsschlüsseln entweder auf dem Master-VDA-Image oder den physischen VDA-Maschinen:

- Für 32-Bit-Betriebssysteme: HKLM\Software\Citrix\GroupPolicy\Defaults\Deprecated
- Für 64-Bit-Betriebssysteme: HKLM\Software\WOW6432Node\Citrix\GroupPolicy\Defaults\Deprecated

Zum Aktivieren der COM-Port- und LPT-Portumleitung, fügen Sie neue Registrierungsschlüssel vom Typ REG_DWORD wie folgt hinzu:

Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab.

Registrierungsschlüssel	Beschreibung	Zulässige Werte
AllowComPortRedirection	Zulassen oder Verhindern der COM-Portumleitung	1 (Zulassen) oder 0 (Verhindern)
LimitComBw	Bandbreitenlimit für COM-Portumleitungskanal	Numerischer Wert
LimitComBWPercent	Bandbreitenlimit für COM-Portumleitungskanal als Prozentsatz der Gesamtsitzungsbandbreite	Numerischer Wert zwischen 0 und 100
AutoConnectClientComPorts	Automatische Verbindung von COM-Ports auf dem Benutzergerät	1 (Zulassen) oder 0 (Verhindern)
AllowLptPortRedirection	Zulassen oder Verhindern der LPT-Portumleitung	1 (Zulassen) oder 0 (Verhindern)
LimitLptBw	Bandbreitenlimit für LPT-Portumleitungskanal	Numerischer Wert
LimitLptBWPercent	Bandbreitenlimit für LPT-Portumleitungskanal als Prozentsatz der Gesamtsitzungsbandbreite	Numerischer Wert zwischen 0 und 100
AutoConnectClientLptPorts	Automatische Verbindung von LPT-Ports auf dem Benutzergerät	1 (Zulassen) oder 0 (Verhindern)

Registrierungsschlüssel	Beschreibung	Zulässige Werte
	Nach dem Konfigurieren dieser Einstellungen ändern Sie die Maschinenkataloge, damit sie das neue Masterimage oder die aktualisierte physische Maschine verwenden. Wenn sich die Benutzer das nächste Mal abmelden, werden die Desktops mit den neuen Einstellungen aktualisiert.	

Benutzerprofile

Sep 29, 2015

Standardmäßig wird bei der Installation des Virtual Delivery Agents die Citrix Profilverwaltung 5.1 ohne Benutzereingriff auf Masterimages installiert. Sie muss jedoch nicht als Profillösung verwendet werden.

Mit XenApp- und XenDesktop-Richtlinien können Sie auf die Maschinen jeder Bereitstellungsgruppe ein anderes Profilverhalten anwenden, um die Profile an unterschiedliche Benutzerbedürfnisse anzupassen. Beispiel: Eine Bereitstellungsgruppe erfordert möglicherweise verbindliche Citrix Profile, deren Vorlage an einem Netzwerkspeicherort gespeichert ist, aber eine andere Bereitstellungsgruppe erfordert möglicherweise Citrix Roamingprofile an einem anderen Speicherort mit mehreren umgeleiteten Ordnern.

Unabhängig davon, für welche Lösung Sie sich entscheiden, können Director-Administratoren auf Diagnoseinformationen für Benutzerprofile zugreifen und die Problembehandlung durchführen. Weitere Informationen finden Sie in der Dokumentation für Director.

Wichtige Informationen zu Richtlinien

Wenn andere Administratoren in Ihrer Organisation für XenApp- und XenDesktop-Richtlinien zuständig sind, stimmen Sie gemeinsam ab, welche profilbezogenen Richtlinien für die Bereitstellungsgruppen gelten.

Richtlinien zur Profilverwaltung können auch in der Gruppenrichtlinie, in der INI-Datei der Profilverwaltung und lokal auf einzelnen virtuellen Maschinen festgelegt werden. Diese verschiedenen Methoden zum Definieren des Profilverhaltens werden in der folgenden Reihenfolge gelesen:

1. Gruppenrichtlinie (ADM- oder ADMX-Dateien)
2. XenApp- und XenDesktop-Richtlinien im Knoten "Richtlinie"
3. Lokale Richtlinien auf der virtuellen Maschine, zu der der Benutzer eine Verbindung herstellt
4. INI-Datei der Profilverwaltung

Beispiel: Wenn Sie die gleiche Richtlinie sowohl in der Gruppenrichtlinie als auch im Knoten "Richtlinie" konfigurieren, wird die Richtlinieneinstellung in der Gruppenrichtlinie vom System gelesen und die XenApp- bzw. XenDesktop-Richtlinieneinstellung wird ignoriert.

Profilverwaltung und Personal vDisk

Wenn Sie das Personal vDisk-Feature verwenden, werden Citrix Benutzerprofile standardmäßig auf den persönlichen vDisks der virtuellen Desktops gespeichert. Löschen Sie die Kopie eines Profils im Benutzerspeicher nicht, wenn gleichzeitig eine Kopie auf der persönlichen vDisk verbleibt. Dies würde einen Profilverwaltungsfehler auslösen und dazu führen, dass für Anmeldungen an dem virtuellen Desktop ein temporäres Profil verwendet wird.

Funktionsweise der automatischen Konfiguration

Der Desktoptyp wird automatisch basierend auf der VDA-Installation erkannt und entsprechende Standardwerte für die Profilverwaltung werden neben Ihrer Konfigurationsauswahl in Studio festgelegt.

Die Richtlinien, die von der Profilverwaltung angepasst werden, werden in der Tabelle angezeigt. Nicht-Standard-Richtlinieneinstellungen bleiben erhalten und werden nicht von diesem Feature überschrieben. Weitere Informationen zu jeder Richtlinie finden Sie in der Dokumentation zur Profilverwaltung.

Die Maschinentypen, für die Profile erstellt werden, wirken sich auf die angepassten Richtlinien aus. Wichtig ist, ob Maschinen persistent oder bereitgestellt sind, und ob sie von mehreren Benutzern gemeinsam verwendet werden oder nur einem dedizierten Benutzer zugeordnet sind.

Persistente Systeme verfügen über einen lokalen Speicher, dessen Inhalt auch nach dem Abschalten des Systems bestehen bleibt. Persistente Systeme imitieren u. U. mit Speichertechnologien wie Speichernetzwerke (Storage Area Networks, SANs) einen lokalen Datenträger. Bereitgestellte Systeme werden dagegen bei Bedarf von einem Basisdatenträger und einem Identitätsdatenträger erstellt. Der lokale Speicher wird üblicherweise durch eine RAM-Disk oder Netzwerk-Disk imitiert. Letztere wird oft über ein SAN mit einer Hochgeschwindigkeitsverbindung zur Verfügung gestellt. Für die Bereitstellung wird allgemein Provisioning Services oder die Maschinenerstellungsdienste (oder ein entsprechendes Produkt eines Drittanbieters) verwendet. Manchmal haben bereitgestellte Systeme persistenten lokalen Speicher, der möglicherweise durch persönliche vDisks zur Verfügung gestellt wird; diese werden als persistent klassifiziert.

Zusammen definieren diese beiden Faktoren die folgenden Maschinentypen:

- **Persistent und dediziert:** Beispiele sind Desktopbetriebssystemmaschinen mit festen Zuweisungen und einer persönlichen vDisk, die mit den Maschinenerstellungsdiensten erstellt wurden, Desktops mit persönlichen vDisks, die in VDI-in-a-Box erstellt wurden, physische Arbeitsstationen und Laptops.
- **Persistent und freigegeben:** Beispiele sind Serverbetriebssystemmaschinen, die mit den Maschinenerstellungsdiensten erstellt wurden.
- **Bereitgestellt und dediziert:** Beispiele sind Desktopbetriebssystemmaschinen mit statischen Zuweisungen, aber ohne persönliche vDisk, die mit Provisioning Services erstellt wurden.
- **Bereitgestellt und freigegeben:** Beispiele sind Desktopbetriebssystemmaschinen mit zufälliger Zuweisung, die mit Provisioning Services erstellt wurden, und Desktops ohne persönliche vDisks, die mit VDI-in-a-Box erstellt wurden.

Die folgenden Richtlinieneinstellungen der Profilverwaltung werden für die verschiedenen Maschinentypen empfohlen. Sie funktionieren in den meisten Fällen gut, aber Sie müssen sie ggf. an die Anforderungen Ihrer Bereitstellung anpassen.

Wichtig: Lokal zwischengespeicherte Profile nach Abmeldung löschen, Profilstreaming und Immer zwischenspeichern werden durch die automatische Konfiguration erzwungen. Passen Sie die anderen Richtlinien manuell an.

Richtlinie	Persistent und dediziert	Persistent und freigegeben	Bereitgestellt und dediziert	Bereitgestellt und freigegeben
Lokal zwischengespeicherte Profile nach Abmeldung löschen	Deaktiviert	Aktiviert	Deaktiviert (Hinweis 5)	Aktiviert
Profilstreaming	Deaktiviert	Aktiviert	Aktiviert	Aktiviert
Immer zwischenspeichern	Aktiviert (Hinweis 1)	Deaktiviert (Hinweis 2)	Deaktiviert (Hinweis 6)	Deaktiviert
Aktiv zurückschreiben	Deaktiviert	Deaktiviert (Hinweis 3)	Aktiviert	Aktiviert
Anmeldungen lokaler Administratoren verarbeiten	Aktiviert	Deaktiviert (Hinweis 4)	Aktiviert	Aktiviert (Hinweis 7)

Richtlinie	Persistent und dediziert	Persistent und freigegeben	Bereitgestellt und dediziert	Bereitgestellt und freigegeben
Hinweise				

1. Da Profilstreaming für diesen Maschinentyp deaktiviert ist, wird die Einstellung Immer zwischenspeichern immer ignoriert.
2. Deaktivieren Sie Immer zwischenspeichern. Sie können sicherstellen, dass große Dateien möglichst bald nach der Anmeldung in Profile geladen werden, wenn Sie diese Richtlinie aktivieren und mit ihr ein Dateigrößenlimit definieren (in MB). Dateien, die diese Größe überschreiten, werden so schnell wie möglich lokal zwischengespeichert.
3. Deaktivieren Sie Aktiv zurückschreiben, außer wenn Sie Änderungen in Profilen für Benutzer speichern, die zwischen XenApp-Servern roamen. Aktivieren Sie in dieser Situation diese Richtlinie.
4. Deaktivieren Sie Anmeldungen lokaler Administratoren verarbeiten, außer für gehostete, freigegebene Desktops. Aktivieren Sie in dieser Situation diese Richtlinie.
5. Deaktivieren Sie Lokal zwischengespeicherte Profile nach Abmeldung löschen. Damit bleiben lokal zwischengespeicherte Profile erhalten. Da die Maschinen beim Abmelden zurückgesetzt werden, aber einzelnen Benutzern zugewiesen sind, ist die Anmeldung mit zwischengespeicherten Profile schneller.
6. Deaktivieren Sie Immer zwischenspeichern. Sie können sicherstellen, dass große Dateien möglichst bald nach der Anmeldung in Profile geladen werden, wenn Sie diese Richtlinie aktivieren und mit ihr ein Dateigrößenlimit definieren (in MB). Dateien, die diese Größe überschreiten, werden so schnell wie möglich lokal zwischengespeichert.
7. Aktivieren Sie Anmeldungen lokaler Administratoren verarbeiten, außer für Benutzer, die zwischen XenApp-Servern roamen. Deaktivieren Sie in dieser Situation diese Richtlinie.

Ordnerumleitung

Hinweis: Konfigurieren Sie die Ordnerumleitung entweder mit den Citrix Richtlinien oder den Active Directory-Gruppenrichtlinienobjekten, jedoch nicht mit beiden. Das Konfigurieren der Ordnerumleitung mit beiden Richtlinienengines kann zu unvorhersehbarem Verhalten führen.

Sie können die Ordnerumleitung mit Richtlinien in Studio konfigurieren. Die Ordnerumleitung ermöglicht das Speichern von Benutzerdaten auf Netzwerkfreigaben, die nicht zum Speichern von Profilen verwendet werden. Dies verringert die Profilgröße und die Ladezeit, hat aber möglicherweise Auswirkungen auf die Netzwerkbandbreite.

Zur Ordnerumleitung müssen keine Citrix Benutzerprofile verwendet werden. Sie können die Benutzerprofile selbst verwalten und dennoch Ordner umleiten.

Allgemeine Empfehlungen für die Ordnerumleitung

Achten Sie beim Einrichten der Ordnerumleitung in Studio auf Folgendes:

- Stellen Sie sicher, dass die Netzwerkspeicherorte zum Speichern des Inhalts von umgeleiteten Ordnern verfügbar sind, und die erforderlichen Berechtigungen richtig sind. Die Speicherorteigenschaften werden überprüft.
- Umgeleitete Ordner werden im Netzwerk eingerichtet und mit Inhalten der virtuellen Desktops bei der Anmeldung aufgefüllt.

Erweiterte Ordnerumleitung

In Bereitstellungen mit mehreren Betriebssystemen können Sie Teile eines Benutzerprofils für jedes Betriebssystem freigeben. Der Rest des Profils ist nicht freigegeben und kann nur von einem Betriebssystem verwendet werden. Um sicherzustellen, dass die Benutzererfahrung für alle Betriebssysteme konsistent ist, benötigen Sie für jedes Betriebssystem eine andere Konfiguration. Dies ist die erweiterte Ordnerumleitung. Beispiel: Bei verschiedenen Versionen einer Anwendung auf zwei Betriebssystemen muss möglicherweise eine freigegebene Datei gelesen oder bearbeitet werden. Sie entscheiden

daher, sie an einen einzigen Speicherort im Netzwerk umzuleiten, von dem beide Versionen auf sie zugreifen können. Oder Sie entscheiden, da die Inhalte des Startmenüordners der beiden Betriebssysteme unterschiedlich strukturiert sind, nur einen Ordner statt beide umzuleiten. Dadurch werden die Startmenüordner und die Inhalte auf jedem Betriebssystem getrennt, und die Benutzererfahrung ist konsistent.

Wenn Sie die erweiterte Ordnerumleitung in Ihrer Bereitstellung benötigen, müssen Sie die Struktur der Profildaten Ihrer Benutzer genau kennen und festlegen, welche Teile davon zwischen Betriebssystemen freigegeben werden können. Dies ist wichtig, weil eine falsch angewendete Ordnerumleitung zu unvorhersehbarem Verhalten führen kann.

Umleiten von Ordnern in erweiterten Bereitstellungen

1. Verwenden Sie eine separate Bereitstellungsgruppe für jedes Betriebssystem.
2. Informieren Sie sich, wo die Benutzerdaten und -einstellungen von den virtuellen Anwendungen, einschließlich solcher auf virtuellen Desktops, gespeichert werden, und wie die Daten strukturiert sind.
3. Leiten Sie die Ordner bei freigegebenen Profildaten, bei denen ein sicheres Datenroaming gewährleistet ist (da sie in jedem Betriebssystem identisch strukturiert sind), in jeder Bereitstellungsgruppe um.
4. Bei nicht freigegebenen Profildaten, für die kein Roaming möglich ist, leiten Sie den Ordner nur in einer Desktopgruppe um. Dies ist in der Regel diejenige mit dem am häufigsten verwendeten Betriebssystem oder diejenige mit den relevantesten Daten. Alternativ können Sie bei nicht freigegebenen Daten, für die kein Roaming zwischen Betriebssystemen möglich ist, die Ordner beider Betriebssysteme an separate Netzwerkadressen umleiten.

Beispiel einer erweiterten Bereitstellung

In diesem Beispiel werden Anwendungen, einschließlich Versionen von Microsoft Outlook und Internet Explorer auf Windows 8-Desktops und Anwendungen sowie andere Versionen von Outlook und Internet Explorer von Windows Server 2008 bereitgestellt. Sie haben hierfür bereits zwei Bereitstellungsgruppen für die beiden Betriebssysteme eingerichtet. Die Benutzer möchten auf dieselben Kontakte und Favoriten in beiden Versionen dieser beiden Anwendungen zugreifen.

Wichtig: Die folgenden Entscheidungen und Hinweise gelten für die hier beschriebenen Betriebssysteme und die beschriebene Bereitstellung. Die Ordner, die Sie in Ihrer Organisation umleiten oder freigeben, hängen von mehreren Faktoren ab, die nur für Ihre Bereitstellung relevant sind.

Sie leiten mit Richtlinien, die auf Bereitstellungsgruppen angewendet werden, die folgenden Ordner um:

Ordner	Umleitung in Windows 8?	Umleitung in Windows Server 2008?
Dokumente	Ja	Ja
Anwendungsdaten	Nein	Nein
Kontakte	Ja	Ja
Desktop	Ja	Nein
Downloads	Nein	Nein
Favoriten	Ja	Ja

Ordner	Umleitung in Windows 8?	Umleitung in Windows Server 2008?
Verknüpfungen	Ja	Nein
Musik	Ja	Ja
Bilder	Ja	Ja
Videos	Ja	Ja
Suchen	Ja	Nein
Gespeicherte Spiele	Nein	Nein
Startmenü	Ja	Nein

Beachten Sie die folgenden Informationen zu freigegebenen, umgeleiteten Ordnern:

- Nach der Analyse der Datenstruktur der von anderen Versionen von Outlook und Internet Explorer gespeicherten Daten entscheiden Sie, dass es sicher ist, die Ordner für Kontakte und Favoriten freizugeben.
- Sie wissen, dass die Struktur der Ordner Eigene Dokumente, Musik, Bilder und Videos auf allen Betriebssystemen gleich ist, und sie können daher für jede Bereitstellungsgruppe am gleichen Netzwerkspeicherort gespeichert werden.

Beachten Sie die folgenden Informationen zu nicht freigegebenen, umgeleiteten Ordnern:

- Die Ordner Desktop, Verknüpfungen, Suchen oder Startmenü werden nicht in die Windows Server-Bereitstellungsgruppe umgeleitet, da die Daten dieser Ordner unterschiedlich in den beiden Betriebssystemen organisiert sind. Eine Freigabe ist daher nicht möglich.
- Um ein vorhersagbares Verhalten für diese nicht freigegebenen Daten sicherzustellen, leiten Sie sie nur in der Windows 8-Bereitstellungsgruppe um. Sie entscheiden dies, da Windows 8 öfter von den Benutzern bei ihrer täglichen Arbeit verwendet wird, die vom Server bereitgestellten Anwendungen hingegen werden nur gelegentlich genutzt. Außerdem sind in diesem Fall die nicht freigegebenen Daten relevanter für eine Desktop- als für eine Anwendungsumgebung. Desktopverknüpfungen werden beispielsweise im Ordner "Desktop" gespeichert und sind nützlich, wenn sie von einer Windows 8-Maschine, aber nicht von einem Windows-Server stammen.

Beachten Sie die folgenden Informationen zu nicht umgeleiteten Ordnern:

- Die Server sollen keine von Benutzern heruntergeladene Dateien ansammeln, und Sie leiten den Ordner "Downloads" daher nicht um.
- Daten von einzelnen Anwendungen können zu Kompatibilitäts- und Leistungsproblemen führen. Daher leiten Sie den Ordner "Anwendungsdaten" nicht um.

Weitere Informationen zur Ordnerumleitung finden Sie unter <http://technet.microsoft.com/en-us/library/cc766489%28v=ws.10%29.aspx>.

Ordnerumleitung und Ausschlüsse

In der Citrix Profilverwaltung (nicht aber in Studio) können Sie mit einer Leistungsverbesserung die Ordnerverarbeitung mit Ausschlüssen verhindern. Wenn Sie dieses Feature verwenden, schließen Sie keine umgeleiteten Ordner aus. Die Ordnerumleitung und die Ausschlussfunktion arbeiten zusammen, d. h. wenn Sie sicherstellen, dass keine umgeleiteten Ordner ausgeschlossen sind, können sie von der Profilverwaltung wieder in die Profildruckerstruktur verschoben werden, während gleichzeitig die Datenintegrität beibehalten wird, sollten sie später doch nicht umgeleitet werden.

Erhalten der Sitzungsaktivität

Sep 29, 2015

Das Erhalten der Sitzungsaktivität ist wichtig, um eine optimale Benutzererfahrung zu gewährleisten. Eine Unterbrechung der Verbindung aufgrund von unzuverlässigen Netzwerken, stark variierender Netzwerklatenz oder Bereichseinschränkungen von drahtlosen Geräten kann zu Frustrationen bei den Benutzern führen. Ein schneller Wechsel zwischen Arbeitsstationen und Zugriff auf dieselben Anwendungen bei jeder Anmeldung ist eine Priorität für viele mobile Mitarbeiter, z. B. von Mitarbeitern in einem Krankenhaus.

Die folgenden Features dienen dazu, die Sitzungszuverlässigkeit zu optimieren, Unannehmlichkeiten und Ausfallzeiten zu reduzieren, Produktivitätsverluste durch unterbrochene Netzwerkkonnektivität zu minimieren und mobilen Benutzern einen schnellen und einfachen Wechsel zwischen Clientgeräten zu ermöglichen:

- Sitzungszuverlässigkeit
- Automatische Wiederverbindung von Clients
- ICA-Keep-Alive
- Workspace Control

Konfigurieren der Sitzungszuverlässigkeit

Durch die Sitzungszuverlässigkeit bleiben Sitzungen aktiv und auf dem Bildschirm des Benutzers, wenn die Netzwerkkonnektivität unterbrochen wird. Die Benutzer sehen so lange weiterhin die Anwendung, die sie verwenden, bis die Netzwerkkonnektivität wiederhergestellt ist.

Diese Funktion ist besonders für mobile Benutzer mit drahtlosen Verbindungen geeignet. Ein Benutzer mit einer drahtlosen Verbindung fährt z. B. in einen Tunnel und die Verbindung wird vorübergehend unterbrochen. Normalerweise würde die Sitzung getrennt und nicht mehr auf dem Bildschirm angezeigt. Der Benutzer müsste sich neu mit der getrennten Sitzung verbinden.

Mit der Sitzungszuverlässigkeit bleibt die Sitzung auf der Maschine aktiv. Der Verlust der Verbindung ist zu erkennen, da auf dem Client der Bildschirm fixiert und der Mauszeiger als sich drehende Sanduhr angezeigt wird, bis die Verbindung am Ende des Tunnels wiederhergestellt ist. Der Benutzer kann während der Unterbrechung weiterhin auf die Anzeige zugreifen und mit der Anwendung weiterarbeiten, wenn die Netzwerkverbindung wiederhergestellt ist. Die Sitzungszuverlässigkeit verbindet Benutzer ohne Neuauthentifizierung wieder.

Citrix Receiver-Benutzer können die Controllereinstellung nicht überschreiben.

Sie können die Sitzungszuverlässigkeit mit SSL (Secure Sockets Layer) verwenden.

Hinweis: Mit SSL werden nur die Daten verschlüsselt, die zwischen dem Benutzergerät und NetScaler Gateway gesendet werden.

Die Sitzungszuverlässigkeit ist standardmäßig über Richtlinieneinstellungen aktiviert. Sie können den für eingehende Sitzungszuverlässigkeitsverbindungen verwendeten Port ändern und die Dauer bearbeiten, für die die Sitzungszuverlässigkeit die Verbindung mit einer unterbrochenen Sitzung aufrechterhält.

Mit der Richtlinieneinstellung Sitzungszuverlässigkeit - Verbindungen können Sie die Sitzungszuverlässigkeit aktivieren oder deaktivieren.

Der Standardwert für die Einstellung Sitzungszuverlässigkeit - Timeout ist 180 Sekunden (3 Minuten). Obwohl Sie den Zeitraum vergrößern können, den die Sitzungszuverlässigkeit eine Sitzung offen lässt, sollten Sie dabei berücksichtigen, dass

diese Funktion den Benutzer nicht zu einer Neuauthentifizierung auffordert, um den Bedienungskomfort zu erhöhen. Je länger eine Sitzung offen gelassen wird, desto höher ist das Risiko, dass der Benutzer abgelenkt wird und das Benutzergerät verlässt. Benutzer ohne Berechtigung hätten in dem Fall möglicherweise Zugriff auf die Sitzung.

Eingehende Sitzungszuverlässigkeitsverbindungen verwenden Port 2598, es sei denn, die Portnummer wurde unter Sitzungszuverlässigkeit - Portnummer geändert.

Verwenden Sie die Funktion zur automatischen Wiederverbindung von Clients, wenn Sie möchten, dass Benutzer eine Verbindung mit unterbrochenen Sitzungen nur mit einer Neuauthentifizierung wiederherstellen können. Sie können die Einstellung für die Citrix Richtlinie Authentifizierung bei automatischer Wiederverbindung von Clients so konfigurieren, dass Benutzer aufgefordert werden, sich neu zu authentifizieren, wenn sie sich mit einer unterbrochenen Sitzung wiederverbinden.

Wenn Sie sowohl Sitzungszuverlässigkeit als auch die Funktion zur automatischen Wiederverbindung verwenden, werden beide Funktionen nacheinander ausgeführt. Die Sitzungszuverlässigkeit beendet oder trennt die Benutzersitzung, nachdem der mit der Option Sitzungszuverlässigkeit - Timeout festgelegte Zeitraum abgelaufen ist. Anschließend werden die Richtlinieneinstellungen für die automatische Wiederverbindung von Clients wirksam und es wird versucht, eine Verbindung mit der unterbrochenen Sitzung wiederherzustellen.

Konfigurieren der automatischen Wiederverbindung von Clients

Benutzer können von ICA-Sitzungen aufgrund von unzuverlässigen Netzwerken, stark variierender Netzwerklatenz oder Bereichseinschränkungen von drahtlosen Geräten getrennt werden. Mit der automatischen Wiederverbindung von Clients kann Receiver unabsichtlich getrennte ICA-Sitzungen erkennen und die Benutzer automatisch wieder mit den betroffenen Sitzungen verbinden. Wenn diese Funktion auf dem Server aktiviert ist, müssen Benutzer nicht manuell eine neue Verbindung herstellen, um mit ihrer Arbeit fortfahren zu können. Receiver versucht, die Verbindung mit der Sitzung wiederherzustellen, bis die Wiederverbindung erfolgreich war oder der Benutzer die Wiederverbindung abbricht.

Konfigurieren Sie die automatische Wiederverbindung mit den folgenden Richtlinieneinstellungen:

- Automatische Wiederverbindung von Clients: Aktiviert oder deaktiviert die automatische Wiederverbindung desselben Clients, nach dem die Verbindung unterbrochen wurde.
- Authentifizierung bei automatischer Wiederverbindung von Clients: Aktiviert oder deaktiviert die erforderliche Benutzerauthentifizierung bei der automatischen Wiederverbindung
- Protokollierung der automatischen Wiederverbindung von Clients: Aktiviert oder deaktiviert die Protokollierung von Wiederverbindungsereignissen im Ereignisprotokoll. Die Protokollierung ist standardmäßig deaktiviert. Wenn diese Einstellung aktiviert ist, werden Informationen zu erfolgreichen oder fehlgeschlagenen automatischen Wiederverbindungsereignissen im Systemprotokoll des Servers aufgezeichnet. Jeder Server speichert die Informationen zu Wiederverbindungsereignissen im eigenen Systemprotokoll; die Site stellt kein Protokoll aller Wiederverbindungsereignisse aller Server bereit.

Bei der automatischen Wiederverbindung von Clients findet eine Authentifizierung mit verschlüsselten Anmeldeinformationen statt. Wenn sich ein Benutzer an einer Site anmeldet, verschlüsselt und speichert der Server die Anmeldeinformationen und erstellt und sendet ein Cookie mit einem Verschlüsselungsschlüssel an Receiver. Receiver übergibt den Schlüssel zum Wiederverbinden an den Server. Der Server entschlüsselt die Anmeldeinformationen und gibt sie an die Windows-Anmeldung für eine Authentifizierung weiter. Benutzer müssen sich beim Ablauf von Cookies neu authentifizieren, um Sitzungen wiederherzustellen.

Cookies werden nicht verwendet, wenn Sie die Einstellung Authentifizierung bei automatischer Wiederverbindung von Clients aktivieren. Stattdessen wird der Benutzer in einem Dialogfeld zur Eingabe der Anmeldeinformationen aufgefordert,

wenn Receiver versucht, die Verbindung automatisch wiederherzustellen.

Hinweis: Für einen maximalen Schutz der Anmeldeinformationen von Benutzern und von Sitzungen sollten Sie die SSL-Verschlüsselung für die gesamte Kommunikation zwischen Clients und Site verwenden.

Sie deaktivieren die automatische Wiederverbindung in Citrix Receiver für Windows mit der Datei `icaclient.adm`. Diese Datei liegt derzeit nur in Englisch vor. Weitere Informationen finden Sie in der Dokumentation von

— *Receiver für Windows*
in den eDocs.

Einstellungen für Verbindungen wirken sich auch auf die automatische Wiederverbindung von Clients aus.

Konfigurieren der Verbindungen für die automatische Wiederverbindung von Clients

In der Standardeinstellung wird die automatische Wiederverbindung von Clients durch Richtlinieneinstellungen auf der Siteebene aktiviert. Der Benutzer muss sich nicht authentifizieren. Wenn jedoch die ICA-TCP-Verbindung eines Servers so konfiguriert wurde, dass Sitzungen mit einer unterbrochenen Kommunikationsverbindung zurückgesetzt werden, findet die automatische Wiederverbindung nicht statt. Die automatische Wiederverbindung von Clients funktioniert nur, wenn der Server Sitzungen trennt, wenn eine unterbrochene Verbindung oder eine Verbindungszeitout vorliegt.

In diesem Zusammenhang bezieht sich die ICA-TCP-Verbindung auf den virtuellen Serverport (nicht auf eine tatsächliche Netzwerkverbindung), der für Sitzungen in TCP/IP-Netzwerken verwendet wird.

Standardmäßig ist die ICA-TCP-Verbindung auf einem Server so eingestellt, dass Sitzungen mit unterbrochenen Verbindungen oder Verbindungen, die das Zeitlimit überschritten haben, getrennt werden. Getrennte Sitzungen bleiben im Systemspeicher intakt und stehen für eine Wiederverbindung durch Receiver zur Verfügung.

Die Verbindung kann so konfiguriert werden, dass Sitzungen mit unterbrochenen Verbindungen oder Verbindungen mit Timeouts zurückgesetzt oder abgemeldet werden. Wenn eine Sitzung zurückgesetzt wird, wird bei einem Wiederverbindungsversuch eine neue Sitzung eingeleitet; die Umgebung des Benutzers wird in der verwendeten Anwendung nicht wiederhergestellt, sondern die Anwendung wird neu gestartet.

Wenn der Server für das Zurücksetzen von Sitzungen konfiguriert ist, erstellt die automatische Wiederverbindung von Clients eine neue Sitzung. Benutzer müssen dann ihre Anmeldeinformationen eingeben, um sich am Server anzumelden.

Die automatische Wiederverbindung kann fehlschlagen, wenn Receiver oder das Plug-In falsche Authentifizierungsinformationen übergeben (dies kann während eines Angriffs passieren), oder wenn der Server feststellt, dass zu viel Zeit seit dem Erkennen der unterbrochenen Verbindung verstrichen ist.

Konfigurieren von ICA-Keep-Alive

ICA-Keep-Alive verhindert, dass Sitzungen durch unterbrochene Verbindungen getrennt werden. Wenn der Server keine Aktivität erkennt (z. B. keine Zeitänderungen, Mausbewegungen oder Bildschirmaktualisierungen) wird verhindert, dass die Sitzung durch die Remotedesktopdienste getrennt wird. Der Server sendet alle paar Sekunden Keep-Alive-Pakete, um zu erkennen, ob die Sitzung aktiv ist. Wenn die Sitzung nicht mehr aktiv ist, wird die Sitzung vom Server als "Getrennt" gekennzeichnet.

Hinweis: ICA-Keep-Alive funktioniert nur, wenn Sie die Sitzungszuverlässigkeit nicht verwenden. Die Sitzungszuverlässigkeit hat eigene Mechanismen für das Aufrechterhalten von Verbindungen. Konfigurieren Sie ICA-Keep-Alive nur für Verbindungen, die keine Sitzungszuverlässigkeit verwenden.

ICA-Keep-Alive-Einstellungen überschreiben Keep-Alive-Einstellungen, die für Microsoft Windows-Gruppenrichtlinien konfiguriert wurden.

Konfigurieren Sie ICA-Keep-Alive mit den folgenden Einstellungen:

- **ICA-Keep-Alive - Timeout:** Gibt das Intervall (1-3600 Sekunden) für das Senden von ICA-Keep-Alive-Meldungen an. Konfigurieren Sie diese Option nicht, wenn die Netzwerksoftware inaktive Sitzungen schließen soll und unterbrochene Verbindungen in der Umgebung so selten sind, dass die Wiederverbindung mit Sitzungen nicht wichtig ist. Die Standardeinstellung von 60 Sekunden bedeutet, dass alle 60 Sekunden ICA-Keep-Alive-Pakete an Benutzergeräte gesendet werden. Antwortet ein Benutzergerät nicht in 60 Sekunden, wird der Status der ICA-Verbindung auf "Getrennt" gesetzt.
- **ICA-Keep-Alives:** Sendet oder verhindert das Senden von ICA-Keep-Alive-Meldungen in gewissen Abständen.

Konfigurieren von Workspace Control

Mit Workspace Control können Desktops und Anwendungen einem Benutzer von einem Gerät zum anderen folgen. Diese Roamingfähigkeit ermöglicht Benutzern den Zugriff auf alle Desktops oder offene Anwendungen von einem beliebigen Ort aus, ohne Neustart des Desktops oder der Anwendungen auf jedem einzelnen Gerät. Sie müssen sich lediglich anmelden. Mit Workspace Control kann das Pflegepersonal in einem Krankenhaus beispielsweise schnell an eine andere Arbeitsstation wechseln und nach der Anmeldung auf dieselben Anwendungen zugreifen. Bei entsprechender Konfiguration von Workspace Control können die Mitarbeiter die Verbindung zu mehreren Anwendungen auf einem Clientgerät trennen und die Verbindung zu denselben Anwendungen auf einem anderen Clientgerät wiederherstellen.

Workspace Control wirkt sich auf die folgenden Aktivitäten aus:

- **Anmelden:** Standardmäßig ermöglicht Workspace Control den Benutzern die Verbindung zu allen ausgeführten Desktops und Anwendungen bei der Anmeldung automatisch wiederherzustellen, ohne sie erneut manuell zu öffnen. Mit Workspace Control können Benutzer getrennte Desktops oder Anwendungen öffnen sowie alle, die auf einem anderen Clientgerät aktiv sind. Beim Trennen der Verbindung mit einem Desktop bzw. einer Anwendung wird das Desktop bzw. die Anwendung weiterhin auf dem Server ausgeführt. Bei Benutzern im Roamingbetrieb, die einige Desktops oder Anwendungen auf einem Clientgerät ausführen müssen, während sie auf einem anderen Clientgerät eine Wiederverbindung zu einem Teil ihres Desktops bzw. ihrer Anwendungen durchführen möchten, können Sie das Wiederverbindungsverhalten bei der Anmeldung so konfigurieren, dass nur die Desktops bzw. Anwendungen geöffnet werden, die zuvor getrennt wurden.
- **Wiederverbinden:** Nach der Anmeldung am Server können Benutzer eine Verbindung zu allen ihren Desktops oder Anwendungen jederzeit wiederherstellen, indem Sie auf "Wiederverbinden" klicken. Beim Wiederverbinden werden standardmäßig sowohl getrennte Desktops oder Anwendungen geöffnet als auch alle aktiven Anwendungen, die derzeit auf einem anderen Clientgerät ausgeführt werden. Sie können die Wiederverbindung so konfigurieren, dass nur die Desktops oder Anwendungen geöffnet werden, deren Verbindung der Benutzer zuvor getrennt hat.
- **Abmelden:** Bei Benutzern, die Desktops oder Anwendungen über StoreFront öffnen, können Sie den Abmeldebefehl so konfigurieren, dass Benutzer entweder von StoreFront und allen aktiven Sitzungen gleichzeitig oder nur von StoreFront abgemeldet werden.
- **Verbindung wird getrennt:** Benutzer können die Verbindung mit allen ausgeführten Desktops und Anwendungen gleichzeitig trennen.

Workspace Control ist nur für Benutzer von Receiver verfügbar, die über eine Citrix StoreFront-Verbindung auf Desktops und Anwendungen zugreifen. Workspace Control ist standardmäßig für virtuelle Desktopsitzungen deaktiviert, für gehostete Anwendungen aber aktiviert. Die Sitzungsfreigabe zwischen veröffentlichten Desktops und veröffentlichten Anwendungen in diesen Desktops erfolgt nicht standardmäßig.

Benutzerrichtlinien, Clientlaufwerkzuordnungen und Druckerkonfigurationen ändern sich entsprechend, wenn ein Benutzer ein neues Clientgerät verwendet. Diese Richtlinien und Zuordnungen werden auf dem Clientgerät angewendet, auf dem der Client derzeit bei der Sitzung angemeldet ist. Wenn sich Pflegepersonal z. B. von einem Clientgerät in der Notaufnahme des Krankenhauses abmeldet und dann an einer Arbeitsstation in der Röntgenabteilung anmeldet, gelten für die Sitzung die Richtlinien, Druckerzuordnungen und Clientlaufwerkzuordnungen der Röntgenabteilung, solange die Sitzung gestartet wird.

Sie können die den Benutzern angezeigten Drucker je nach Standort anpassen. Außerdem können Sie steuern, ob Benutzer auf lokalen Druckern drucken können, wie viel Bandbreite bei einer Remoteverbindung verwendet wird sowie andere Aspekte des Druckens.

Weitere Informationen zur Aktivierung und Konfiguration von Workspace Control für Benutzer finden Sie in der StoreFront-Dokumentation in den eDocs.

Personal vDisk 7.x

Sep 29, 2015

Das Personal vDisk-Feature bietet die Einzelimageverwaltung für gepoolte und gestreamte Desktops, während Benutzer Anwendungen installieren und ihre Desktopeinstellungen anpassen können. Im Gegensatz zu traditionellen Virtual Desktop Infrastructure (VDI)-Bereitstellungen mit gepoolten Desktops, bei denen Benutzer Anpassungen und eigene Anwendungen verlieren, wenn der Administrator das Masterimage ändert, werden in Bereitstellungen mit persönlichen vDisks diese Änderungen beibehalten. Dies bedeutet, dass Administratoren die Masterimages auf einfache Weise zentral verwalten können, während Benutzer gleichzeitig von einer individuell angepassten Desktoperfahrung profitieren.

Persönliche vDisks erreichen diese Trennung, indem Sie alle auf der VM des Benutzers vorgenommenen Änderungen an einen separaten Datenträger, die persönliche vDisk, weiterleiten, die mit der VM des Benutzers verknüpft ist. Der Inhalt der persönlichen vDisk wird zur Laufzeit mit dem Inhalt des Masterimages zusammengeführt, um eine einheitliche Erfahrung zu gewährleisten. Auf diese Weise können Benutzer immer noch auf die Anwendungen zugreifen, die der Administrator auf dem Masterimage zur Verfügung gestellt hat.

Persönliche vDisks bestehen aus zwei Teilen, die unterschiedliche Laufwerksbuchstaben verwenden und ungefähr gleich groß sind:

- Benutzerprofil: Dieser Teil enthält Benutzerdaten, Dokumente und das Benutzerprofil. Standardmäßig wird hierfür Laufwerk P: verwendet, Sie können aber einen anderen Laufwerksbuchstaben wählen, wenn Sie einen Maschinenkatalog mit Maschinen erstellen, die persönliche vDisks verwenden. Das verwendete Laufwerk hängt auch von der Einstellung für EnableUserProfileRedirection ab.
- Virtual Hard Disk-Datei (.vhd): Dieser Teil enthält alle anderen Objekte, z. B. Anwendungen, die unter C:\Programme installiert sind. Dieser Teil wird nicht in Windows Explorer angezeigt und benötigt seit Version 5.6.7 keinen Laufwerksbuchstaben.

Mit persönlichen vDisks können Anwendungen auf Abteilungsebene bereitgestellt werden und sie unterstützen auch Anwendungen, die von Benutzern heruntergeladen und installiert wurden, einschließlich solcher, für die Treiber (außer Phase-1-Treiber), Datenbanken und Maschinenverwaltungssoftware erforderlich ist. Wenn die Änderung eines Benutzers mit der Änderung eines Administrators kollidiert, können diese Änderungen mit einer persönlichen vDisk leicht und automatisch abgestimmt werden.

Außerdem können lokal verwaltete Anwendungen (z. B. solche, die von lokalen IT-Abteilungen bereitgestellt werden) auch in der Umgebung des Benutzers bereitgestellt werden. Der Benutzer bemerkt keine Unterschiede bei der Verwendung; mit persönlichen vDisks wird sichergestellt, dass alle Änderungen und alle installierten Anwendungen auf der vDisk gespeichert werden. In Fällen, bei denen eine Anwendung auf einer persönlichen vDisk genau mit einer Anwendung auf einem Masterimage übereinstimmt, wird die Version auf der persönlichen vDisk aus Platzspargründen verworfen, ohne dass der Benutzer den Zugriff auf die Anwendung verliert.

Persönliche vDisks werden physisch auf dem Hypervisor gespeichert, sie müssen sich aber nicht am gleichen Speicherort befinden wie andere auf dem virtuellen Desktop bereitgestellte Datenträger. Dadurch können sich die Kosten für persönlichen vDisk-Speicher verringern.

Wenn Sie während der Site-Erstellung eine Verbindung erstellen, legen Sie Speicherorte für die Datenträger fest, die von den virtuellen Maschinen verwendet werden. Sie können die persönlichen vDisks von den Datenträgern für das Betriebssystem trennen. Jede VM muss Zugriff auf den Speicherort der beiden Datenträger haben. Wenn Sie für beide lokalen Speicher verwenden, muss der Hypervisor in der Lage sein, auf beide zuzugreifen. Um dies zu gewährleisten, bietet Studio nur

kompatible Speicherorte an. Später können Sie auch über "Konfiguration > Hosting" in Studio persönliche vDisks und Speicher dafür vorhandenen Hosts (aber nicht Maschinenkatalogen) hinzufügen.

Erstellen Sie regelmäßig ein Backup der persönlichen vDisks mit der bevorzugten Methode. vDisks sind standardmäßige Volumes in der Speicherebene eines Hypervisors. Sie können genauso wie andere Volumes gesichert werden.

Neue Features in Personal vDisk 7.6.1

Dieses Release enthält folgende Verbesserungen:

- Diese Version von Personal vDisk enthält Leistungsverbesserungen, durch die die zum Anwenden eines Imageupdates auf einen Personal vDisk-Katalog benötigte Zeit verkürzt wird.

Die folgenden Probleme wurden in diesem Release behoben:

- Bei dem Versuch, eine Basis-VM mit einem direkten Upgrade von Microsoft Office 2010 auf Microsoft Office 2013 zu aktualisieren, wurde dem Benutzer ein Konfigurationsfenster gefolgt von der folgenden Fehlermeldung angezeigt: "Error 25004. The product key you entered cannot be used on this machine." Zuvor wurde in diesem Fall empfohlen, Office 2010 auf der Basis-VM zu deinstallieren und danach Office 2013 zu installieren. Jetzt ist es nicht mehr nötig, Office 2010 beim Durchführen eines direkten Upgrades auf der Basis-VM zu deinstallieren (#391225).
- Wenn während eines Imageupdates eine höhere Version von Microsoft .Net auf der persönlichen vDisk des Benutzers gefunden wurde, wurde diese vom Basisimage mit einer niedrigeren Version überschrieben. Dies führte für Benutzer zu Problemen beim Ausführen bestimmter, auf ihrer persönlichen vDisk installierter Anwendungen, die eine höhere Version, z. B. Visual Studio, benötigten (#439009).
- Ein mit Provisioning Services bereitgestellter Datenträger, auf dem Personal vDisk installiert und aktiviert ist, kann nicht zum Erstellen eines nicht-Personal vDisk-Maschinenkatalogs verwendet werden. Diese Einschränkung wurde entfernt (#485189).

Info zu Personal vDisk 7.6

Neue Features in Version 7.6:

- Verbesserte Fehlerbehandlung und Berichterstellung für Personal vDisk. Wenn Sie in Studio PvD-aktivierte Maschinen in einem Katalog anzeigen, können Sie auf der Registerkarte "PvD" den Überwachungsstatus während Imageupdates sowie die geschätzte Abschlusszeit und den Fortschritt verfolgen. Außerdem wurden die Zustandsanzeigen verbessert.
- Ein Überwachungstool für PvD-Imageupdates ist für frühere Versionen auf dem ISO-Medium verfügbar (ISO\Support\Tools\Scripts\PvdTool). Überwachungsfunktionen wurden auch in früheren Releases unterstützt, jedoch sind die Berichterstellungsfunktionen nicht so robust wie im aktuellen Release.
- Im Provisioning Services-Testmodus können Sie Maschinen mit einem aktualisierten Image in einem Testkatalog starten. Nachdem Sie die Stabilität überprüft haben, können Sie die Testversion der persönlichen vDisk auf "Production" hochstufen.
- Mit einer neuen Funktion können Sie das Delta (die Differenz) zwischen zwei Beständen während einer Bestandsaktualisierung berechnen statt sie für jeden PvD-Desktop einzeln berechnen zu müssen. Es sind neue Befehle zum Exportieren und Importieren eines vorherigen Bestands für MCS-Kataloge verfügbar. (Provisioning Services-Master-vDisks haben bereits den vorherigen Bestand.)

Bekannte Probleme aus Version 7.1.3, die in Version 7.6 behoben wurden:

- Unterbrechen des Upgrades einer Personal vDisk-Installation kann dazu führen, dass eine vorhandene Personal vDisk-Installation beschädigt wird. [#424878]
- Ein virtueller Desktop reagiert möglicherweise nicht mehr, wenn die persönliche vDisk über einen langen Zeitraum ausgeführt wird und es zu einem Speicherverlust im nicht-ausgelagerten Speicher kommt. [#473170]

Neue bekannten Probleme in Version 7.6:

- Das Vorhandensein von Antivirenprodukten kann sich darauf auswirken, wie lange das Durchführen einer Bestandsaktualisierung oder eines Updates dauert. Die Leistung kann verbessert werden, wenn Sie CtxPvD.exe und CtxPvDSvc.exe der PROCESS-Ausschlussliste des Antivirenprodukts hinzufügen. Diese Dateien befinden sich im Ordner C:\Programme\Citrix\personal vdisk\bin. [#326735]
- Feste Links zwischen Dateien, die vom Masterimage geerbt wurden, bleiben in persönlichen vDisk-Katalogen nicht erhalten. [#368678]
- Nach dem Upgrade von Office 2010 auf 2013 auf dem Personal vDisk-Masterimage kann Office nicht auf virtuellen Maschinen gestartet werden, da der KMS-Lizenzierungsproduktschlüssel von Office beim Upgrade entfernt wurde. Als Workaround können Sie Office 2010 deinstallieren und Office 2013 auf dem Masterimage neu installieren. [#391225]
- Personal vDisk-Kataloge unterstützen keine VMware Paravirtual SCSI (PVSCSI)-Controller. Verwenden Sie den Standardcontroller, um dieses Problem zu vermeiden. [#394039]
- Für virtuelle Desktops, die mit Personal vDisk Version 5.6.0 erstellt wurden und auf 7 aktualisiert werden, finden Benutzer, die sich an der Master-VM anmelden, nicht alle Dateien in der gepoolten VM. Dieses Problem tritt auf, da ein neues Benutzerprofil erstellt wird, wenn sich Benutzer an der gepoolten VM anmelden. Für dieses Problem gibt es keinen Workaround. [#392459]
- Personal vDisks, die unter Windows 7 ausgeführt werden, können das Feature "Sichern und Wiederherstellen" nicht verwenden, wenn der Windows-Systemschutz aktiviert ist. Wenn der Systemschutz deaktiviert ist, wird ein Backup des Benutzerprofils aber nicht der Datei userdata.v2.vhd erstellt. Citrix empfiehlt, den Systemschutz zu deaktivieren und ein Backup des Benutzerprofils mit dem Feature "Sichern und Wiederherstellen" zu erstellen. [#360582]
- Beim Erstellen einer VHD-Datei auf der Basis-VM mit dem Datenträgerverwaltungstool können Sie die virtuelle Festplatte (VHD) möglicherweise nicht bereitstellen. Um dieses Problem zu umgehen, kopieren Sie die VHD auf das PvD-Volume. [#355576]
- Office 2010-Verknüpfungen bleiben auf virtuellen Desktops erhalten, nachdem diese Software entfernt wurde. Um dieses Problem zu umgehen, löschen Sie die Verknüpfungen. [#402889]
- Wenn Sie Microsoft Hyper-V verwenden, können Sie mit Maschinen mit persönlichen vDisks keinen Katalog erstellen, wenn die Maschinen lokal gespeichert sind und die vDisks auf freigegebenen Clustervolumes (CSVs) gespeichert sind. Die Katalogerstellung schlägt fehl und ein Fehler wird angezeigt. Um dieses Problem zu umgehen, verwenden Sie einen anderen Speicherort für die vDisks. [#423969]
- Wenn Sie sich das erste Mal bei einem virtuellen Desktop anmelden, der von einem Provisioning Services-Katalog erstellt wurde, werden Sie aufgefordert, den Desktop neu zu starten, wenn die persönliche vDisk zurückgesetzt wurde (mit dem Befehl "ctxpvd.exe -s reset"). Um dieses Problem zu lösen, starten Sie den Desktop neu, wenn Sie dazu aufgefordert werden. Es handelt sich hier um eine einmalige Zurücksetzung, die bei erneuter Anmeldung nicht erforderlich ist. [#340186]
- Wenn Sie .NET 4.5 auf einer persönlichen vDisk installieren und .NET 4.0 bei einem späteren Update des Images installiert oder bearbeitet wird, schlagen Anwendungen, die von .NET 4.5 abhängig sind, fehl. Um dieses Problem zu umgehen, stellen Sie .NET 4.5 vom Basisimage aus als Imageupdate bereit.
- Weitere Informationen finden Sie unter
— *Bekannte Probleme*
in der Dokumentation zu XenApp und XenDesktop 7.6.

Info zu Personal vDisk 7.1.3

Bekannte Probleme aus Version 7.1.1, die in Version 7.1.3 behoben wurden:

- Direkte Upgrades von Personal vDisk 5.6.0 auf Personal vDisk 7.x können zu Fehlern bei Personal vDisk führen. [#432992]
- Benutzer können möglicherweise nur zeitweilige Verbindungen zu virtuellen Desktops mit persönlichen vDisks herstellen. [#437203]

- Wenn das Update eines Personal vDisk-Images während der Aktualisierung von Personal vDisk 5.6.5 oder höher auf Personal vDisk 7.0 oder höher unterbrochen wird, können spätere Updates fehlschlagen. [#436145]

Info zu Personal vDisk 7.1.1

Bekannte Probleme aus Version 7.1, die in Version 7.1.1 behoben wurden:

- Nach der Aktualisierung auf Symantec Endpoint Protection 12.1.3 über ein Imageupdate meldet symhelp.exe beschädigte Antivirus-Definitionen. [#423429]
- Personal vDisk kann den Neustart gepoolter Desktops verursachen, wenn der Dienststeuerungs-Manager (services.exe) abstürzt. [#0365351]

Neue bekannten Probleme in Version 7.1.1: Keine

Info zu Personal vDisk 7.1

Neue Features in Version 7.1:

- Sie können Personal vDisk nun auch auf Desktops unter Windows 8.1 verwenden und die Ereignisprotokollierung wurde verbessert.
- Copy-on-Write (CoW) wird in diesem Release nicht mehr unterstützt. Beim Upgrade von Personal vDisk Version 7.0 auf 7.1 gehen alle von CoW verwalteten Datenänderungen verloren. CoW war ein experimentelles Feature in XenDesktop 7 und es war standardmäßig deaktiviert. Wenn Sie es nicht aktiviert haben, sind Sie nicht betroffen.

Bekannte Probleme aus Version 7.0.1, die in Version 7.1 behoben wurden:

- Wenn der Wert des Personal vDisk-Registrierungsschlüssels EnableProfileRedirection auf 1 oder ON eingestellt ist, und Sie den Wert beim Aktualisieren des Images in 0 oder OFF ändern, wird der ganze Personal vDisk-Speicherplatz u. U. den vom Benutzer installierten Anwendungen zugeordnet, sodass kein Platz für Benutzerprofile vorhanden ist, die auf der vDisk bleiben. Wenn die Profilverwaltung für einen Katalog deaktiviert ist und Sie die Profilverwaltung während eines Imageupdates aktivieren, können sich Benutzer u. U. nicht bei ihren virtuellen Desktops anmelden. [#381921]
- Der Desktopdienst protokolliert nicht den richtigen Fehler in der Ereignisanzeige, wenn das Update des Personal vDisk-Bestands fehlschlägt. [#383331]
- Beim Upgrade auf Personal vDisk 7.x bleiben geänderte Regeln nicht erhalten. Dieses Problem wurde für das Upgrade von Version 7.0 auf Version 7.1 behoben. Beim Upgrade von Version 5.6.5 auf Version 7.1 müssen Sie die Regeldatei zuerst speichern und die Regeln nach dem Upgrade erneut anwenden. [#388664]
- Personal vDisks, die unter Windows 8 ausgeführt werden, können keine Anwendungen vom Windows Store installieren. Eine Fehlermeldung gibt an, dass der Kauf nicht abgeschlossen werden konnte. Wenn Sie den Windows Update-Dienst aktivieren, wird das Problem nun behoben. Von Benutzern installierte Anwendungen müssen jedoch nach dem Neustart des Systems neu installiert werden. [#361513]
- Einige symbolische Verknüpfungen fehlen auf gepoolten Windows 7-Desktops mit persönlichen vDisks. Daher werden Symbole, die von Anwendungen unter C:\Benutzer\Alle Benutzer gespeichert werden, nicht im Startmenü angezeigt. [#418710]
- Der Start einer persönlichen vDisk schlägt fehl, wenn ein USN-Journalüberlauf aufgrund zahlreicher Systemänderungen nach einer Bestandsaktualisierung auftritt. [#369846]
- Der Start einer persönlichen vDisk schlägt fehl und der Statuscode 0x20 sowie Fehlercode 0x20000028 werden angezeigt. [#393627]
- Symantec Endpoint Protection 12.1.3 zeigt die Fehlermeldung "Proactive Threat Protection is malfunctioning" an und der Liveupdatestatus dieser Komponente ist nicht verfügbar. [#390204]

Neue bekannte Probleme in Version 7.1: Weitere Informationen finden Sie unter

— *Bekannte Probleme*

in der Dokumentation zu XenDesktop 7.1.

Info über Personal vDisk 7.0.1

Neu in Version 7.0.1: Personal vDisk ist jetzt robuster bei Änderungen der Umgebung. Virtuelle Desktops mit persönlichen vDisks werden jetzt beim Delivery Controller registriert, selbst wenn das Imageupdate fehlschlägt, und das Herunterfahren von unsicheren Systemen setzt die vDisks in einen permanent deaktivierten Zustand. Darüber hinaus können Sie nun während einer Bereitstellung Dateien und Ordner mithilfe von Regeldateien von vDisks ausschließen.

Bekannte Probleme aus Version 5.6.13, die in Version 7.0.1 behoben wurden:

- Änderungen an der Mitgliedschaft einer Gruppe, die von einem Benutzer an einem gepoolten virtuellen Desktop vorgenommen wurden, gehen u. U. nach einem Imageupdate verloren. [#286227]
- Imageupdates schlagen u. U. mit einer Fehlermeldung, dass der Speicherplatz auf dem Datenträger niedrig ist, fehl, selbst wenn die persönliche vDisk genug Speicherplatz hat. [#325125]
- Die Installation einiger Anwendungen schlägt auf virtuellen Desktops mit einer persönlichen vDisk fehl, und eine Meldung weist auf einen erforderlichen Neustart hin. Grund ist eine ausstehende Umbenennung. [#351520]
- Symbolische Links, die im Masterimage erstellt wurden, funktionieren nicht auf virtuellen Desktops mit persönlichen vDisks. [#352585]
- In Umgebungen mit der Citrix Profilverwaltung und Personal vDisk funktionieren Anwendungen, die Benutzerprofile auf einem Systemvolume überprüfen u. U. nicht richtig, wenn die Profilverwaltung aktiviert ist. [#353661]
- Die Bestandsaktualisierung schlägt auf Masterimages fehl, wenn der Bestand größer als 2 GB ist. [#359768]
- Imageupdates schlagen mit einem Fehlercode 112 fehl und persönliche vDisks sind beschädigt, selbst wenn die vDisks ausreichend Speicherplatz für das Update haben. [#363003]
- Das Skript für das Ändern der Größe schlägt für Kataloge fehl, die mehr als 250 Desktops haben. [#363365]
- Benutzerseitige Änderungen einer Umgebungsvariablen gehen bei einer Aktualisierung des Images verloren. [#372295]
- Lokale Benutzer, die auf einem virtuellen Desktop mit einer persönlichen vDisk erstellt wurden, gehen bei einer Aktualisierung des Images verloren. [#377964]
- Der Start einer persönlichen vDisk schlägt u. U. fehl, wenn ein USN-Journalüberlauf aufgrund zahlreicher Systemänderungen nach einer Bestandsaktualisierung auftritt. Erhöhen Sie die USN-Journalgröße im Masterimage auf mindestens 32 MB und aktualisieren Sie das Image, um dieses Problem zu vermeiden. [#369846]
- Ein Problem wurde mit Personal vDisk festgestellt, das ein richtiges Funktionieren der Registrierungsstrukturaktionen des AppSense-Umgebungs-Managers verhindert, wenn AppSense im Ersetzenmodus verwendet wird. Citrix und AppSense arbeiten an der Behebung des Problems, das mit dem Verhalten der RegRestoreKey-API zusammenhängt, wenn Personal vDisk installiert ist. [#0353936]

Release-unabhängige bekannte Probleme

- Wenn eine Anwendung auf einer persönlichen vDisk (PvD) mit einer auf dem Masterimage installierten anderen Anwendung derselben Version verbunden ist, funktioniert die Anwendung auf der PvD nach einem Imageupdate möglicherweise nicht mehr. Dieses Problem tritt auf, wenn die Anwendung vom Masterimage deinstalliert oder auf eine neuere Version aktualisiert wird, da durch diese Aktion die Dateien entfernt werden, die die Anwendung auf der PvD vom Masterimage benötigt. Um dies zu verhindern, lassen Sie die Anwendung mit den Dateien, die von der Anwendung auf der PvD benötigt werden, auf dem Masterimage.
Beispiel: Das Masterimage enthält Office 2007 und ein Benutzer installiert Visio 2007 auf der PvD. Die Office-Anwendungen und Visio funktionieren einwandfrei. Später ersetzt der Administrator Office 2007 durch Office 2010 auf dem Masterimage und aktualisiert anschließend alle betroffenen Maschinen mit dem aktualisierten Image. Visio 2007 funktioniert nicht mehr. Um dies zu verhindern, lassen Sie Office 2007 auf dem Masterimage. [#320915]
- Wenn Personal vDisk verwendet wird, verwenden Sie bei der Bereitstellung von McAfee Virus Scan Enterprise (VSE) die

Version 8.8 Patch 4 oder höher auf einem Masterimage. [#303472]

- Wenn eine Verknüpfung für eine Datei auf dem Masterimage nicht mehr funktioniert, weil das Verknüpfungsziel in PvD umbenannt wurde, erstellen Sie die Verknüpfung neu. [#367602]
- Verwenden Sie keine absoluten bzw. festen Links auf einem Masterimage. [#368678]
- Das mit Windows 7 verfügbare Feature "Sichern und Wiederherstellen" wird auf der persönlichen vDisk nicht unterstützt. [#360582]
- Nach der Anwendung eines aktualisierten Masterimages ist kein Zugriff auf die Konsole für lokale Benutzer und Gruppen möglich oder sie zeigt inkonsistente Daten an. Um das Problem zu lösen, setzen Sie die Benutzerkonten auf der VM zurück. Dazu muss die Sicherheitsstruktur zurückgesetzt werden. Dieses Problem wurde im Release 7.1.2 behoben und löst auch das Problem für VMs in späteren Releases, aber nicht für VMs, die mit einer früheren Version erstellt und dann aktualisiert wurden. [#488044]
- Bei der Verwendung einer gepoolten VM in einer ESX Hypervisor-Umgebung wird Benutzern eine Neustartaufforderung angezeigt, wenn der ausgewählte SCSI-Controllertyp ein "VMware Paravirtual" ist. Um dieses Problem zu umgehen, verwenden Sie den Controllertyp "LSI SCSI". [#394039]
- Nach dem Zurücksetzen von PvD auf einem mit Provisioning Services erstellten Desktop wird Benutzern nach der Anmeldung an der VM u. U. eine Neustartaufforderung angezeigt. Um dieses Problem zu umgehen, starten Sie den Desktop neu. [#340186]
- Benutzer von Windows 8.1-Desktops können sich u. U. nicht an ihren PvDs anmelden. Einem Administrator wird möglicherweise die Meldung "PvD was disabled due to unsafe shutdown" angezeigt und das PvDActivation-Protokoll enthält u. U. die Meldung "Failed to load reg hive [\Device\IvmVhdDisk00000001\CitrixPvD\Settings\RingCube.dat]". Dieses Problem tritt auf, wenn die VM nicht sicher heruntergefahren wird. Sie umgehen das Problem, indem Sie die persönliche vDisk zurücksetzen. [#474071]

Installation und Upgrade

Sep 29, 2015

Personal vDisk 7.x wird seit XenDesktop 5.6 bis zur aktuellen Version unterstützt. Für jede XenDesktop-Version werden in der Dokumentation unter "Systemanforderungen" die unterstützten Betriebssysteme für Virtual Delivery Agents (VDAs) sowie die unterstützten Versionen von Hosts (Virtualisierungsressourcen) und Provisioning Services aufgelistet. Weitere Informationen zu Provisioning Services-Aufgaben finden Sie in der Dokumentation zu Provisioning Services.

Installieren und Aktivieren von PvD

PvD wird automatisch installiert, wenn Sie einen VDA für Desktopbetriebssysteme auf einer Maschine installieren oder aktualisieren. Wenn Sie die PvD-Software nach der Installation des VDAs aktualisieren, verwenden Sie die [hier](#) verfügbare PvD-MSI (Citrix Anmeldeinformationen erforderlich).

Aktivieren von PvD:

- PvD wird automatisch aktiviert, wenn Sie mit den Maschinenerstellungsdiensten (MCS) einen Maschinenkatalog erstellen, dessen Desktopbetriebssystemmaschinen eine persönliche vDisk verwenden.
- Wenn Sie Provisioning Services (PVS) verwenden, wird PvD automatisch aktiviert, wenn Sie während der Erstellung des Masterimages den Bestand aufnehmen oder wenn bei einem automatischen Update der Bestand aktualisiert wird.

Während der VDA-Installation haben Sie die Möglichkeit, PvD zu aktivieren, indem Sie das Kontrollkästchen "Personal vDisk" auf der grafischen Benutzeroberfläche auswählen oder über die Befehlszeilenschnittstelle die Option "/baseimage" angeben. Auch wenn Sie diese Aktion bei der Installation des VDAs (Standard) nicht ausführen, können Sie mit dem gleichen Image PvD-Desktops und Desktops ohne PvD erstellen, da PvD während der Katalogerstellung aktiviert wird.

Hinzufügen von persönlichen vDisks

Sie fügen Hosts persönliche vDisks hinzu, wenn Sie eine Site konfigurieren. Sie können denselben Speicher auf dem Host für VMs und persönliche vDisks verwenden oder Sie können einen anderen Speicher für persönliche vDisks auswählen.

Später können Sie auch persönliche vDisks und ihren Speicher vorhandenen Hosts (Verbindungen), jedoch nicht Maschinenkatalogen, hinzufügen.

1. Wählen Sie im Studio-Navigationsbereich Konfiguration > Hosting.
2. Wählen Sie im Aktionsbereich Persönlichen vDisk-Speicher hinzufügen und geben Sie einen Speicherort an.

Aktualisieren von PvD

Die einfachste Methode, Personal vDisk von einer früheren 7.x-Version zu aktualisieren, ist das Aktualisieren der VDAs für Desktopbetriebssysteme mit der aktuellen XenDesktop-Version. Nehmen Sie anschließend den PvD-Bestand auf.

Sie können auch nur PvD mit der [hier](#) verfügbaren PvD-MSI aktualisieren.

Deinstallieren von PvD

Sie können die PvD-Software mit einer der folgenden beiden Methoden deinstallieren:

- Deinstallieren Sie den VDA. Dabei wird die PvD-Software ebenfalls entfernt.
- Wenn Sie PvD mit der PvD-MSI aktualisiert haben, können Sie es über die Liste der Programme deinstallieren.

Wenn Sie PvD deinstallieren und dieselbe oder eine neuere Version neu installieren, erstellen Sie eine Sicherungskopie des Registrierungsschlüssels HKLM\Software\Citrix\personal vDisk\config, der Konfigurationseinstellungen für die Umgebung

enthält, die sich geändert haben können. Nach der Installation von PVD können Sie die Registrierungswerte, die sich geändert haben, durch einen Vergleich mit der Sicherungskopie zurücksetzen.

Konfiguration und Verwaltung

Sep 29, 2015

In diesem Abschnitt werden Themen erläutert, die beim Konfigurieren und Verwalten einer Personal vDisk (PvD)-Umgebung zu berücksichtigen sind. Darüber hinaus werden Best Practices und Aufgabenbeschreibungen behandelt.

Beachten Sie Folgendes beim Arbeiten in der Windows-Registrierung:

Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab.

Überlegungen zur Größe einer persönlichen vDisk

Die folgenden Faktoren beeinflussen die Größe des PvD-Hauptvolumes:

- **Größe der Anwendungen, die Benutzer auf ihren PvDs installieren**

Bei Neustarts bestimmt PvD den im Anwendungsbereich verbleibenden freien Speicherplatz (UserData.v2.vhd). Wenn dieser Wert unter 10 % fällt, wird der Anwendungsbereich erweitert, indem ungenutzter Speicherplatz des Profilbereichs (standardmäßig der freie Speicherplatz auf dem Laufwerk P:) genutzt wird. Der zum Anwendungsbereich hinzugefügte Speicherplatz beträgt ungefähr 50 % des zusammengefassten freien Speicherplatzes von Anwendungs- und Profilbereich.

Beispiel: Wenn der Anwendungsbereich auf einer 10 GB PvD, der standardmäßig eine Größe von 5 GB hat, den Wert 4,7 GB erreicht und der Profilbereich über 3 GB freien Speicherplatz verfügt, wird der zusätzliche Speicherplatz wie folgt berechnet:

erweiterter Speicherplatz = $(5,0 - 4,7) : 2 + 3,0 : 2 = 1,65$ GB

Der Wert für den zusätzlichen Speicherplatz kann nur ungefähr angegeben werden, da Abstriche für das Speichern von Protokollen und für Mehraufwand gemacht werden müssen. Die Berechnung und mögliche Größenänderung wird bei jedem Neustart ausgeführt.

- **Größe der Benutzerprofile (wenn keine separate Profilverwaltungslösung verwendet wird)**

Zusätzlich zu dem für Anwendungen erforderlichen Speicherplatz muss auf persönlichen vDisks auch genügend Speicherplatz für das Speichern von Benutzerprofilen vorhanden sein. Schließen Sie alle nicht umgeleiteten speziellen Ordner (z. B. Dokumente und Musik) in Ihre Speicherplatzberechnungen ein. Vorhandene Profilgrößen sind in der Systemsteuerung (sysdm.cpl) verfügbar.

Einige Profilumleitungslösungen speichern Stubdateien (Sentineldateien) anstelle echter Profildaten. Zu Beginn kann es aussehen, als würden die Profillösungen keine Daten speichern. Sie nutzen jedoch einen Dateiverzeichniseintrag pro Stubdatei im Dateisystem, was ungefähr 4 KB pro Datei ausmacht. Wenn Sie eine solche Lösung verwenden, schätzen Sie die Größe anhand der echten Profildaten und nicht basierend auf den Stubdateien.

Unternehmensanwendungen zur Dateifreigabe, wie ShareFile und Dropbox, synchronisieren oder laden Daten möglicherweise auf die Profildatenbereiche der persönlichen vDisks von Benutzern herunter. Wenn Sie derartige Anwendungen verwenden, veranschlagen Sie genug Speicherplatz für diese Daten.

- **Mehraufwand durch die virtuelle Festplattenvorlage, die den PvD-Bestand enthält**

Die virtuelle Festplattenvorlage enthält die PvD-Bestandsdaten (Sentineldateien, die zum Inhalt des Masterimages

gehören). Der PvD-Anwendungsbereich wird von dieser virtuellen Festplatte erstellt. Da jede Sentineldatei bzw. jeder Sentinelordner einen Dateiverzeichniseintrag im Dateisystem enthält, nimmt der Inhalt der virtuellen Festplattenvorlage PvD-Anwendungsspeicherplatz in Anspruch, bevor der Endbenutzer überhaupt eine Anwendung installiert hat. Sie können die Größe der virtuellen Festplattenvorlage bestimmen, indem Sie zum Masterimage navigieren, nachdem der Bestand aufgenommen wurde. Sie können zur ungefähren Berechnung auch die folgende Formel verwenden:

Größe der virtuellen Festplattenvorlage = (Anzahl der Dateien des Basisimage) x 4 KB

Sie ermitteln die Anzahl der Dateien und Ordner, indem Sie mit der rechten Maustaste auf das Laufwerk C: des Images der Basis-VM klicken und Eigenschaften auswählen. Beispiel: Ein Image mit 250.000 Dateien ergibt eine virtuelle Festplattenvorlage von ungefähr 1.024.000.000 Bytes (nicht ganz 1 GB). Dieser Speicherplatz ist nicht für Anwendungsinstallationen im PvD-Anwendungsbereich verfügbar.

- **Mehraufwand für PvD-Imageupdatevorgänge**

Während PvD-Imageupdatevorgänge ausgeführt werden, muss im Stammverzeichnis der PvD (standardmäßig P:) genug Speicherplatz vorhanden sein, um die Änderungen der zwei Imageversionen mit den Änderungen, die der Benutzer an der PvD vorgenommen hat, zusammenzuführen. Normalerweise reserviert die PvD einige Hundert MB für diesen Zweck. Durch Extradaten, die im Laufwerk P: gespeichert wurden, ist jedoch möglicherweise nicht genug Speicherplatz zum Durchführen des Imageupdates vorhanden. Mit dem PvD-Poolstatistikscript (auf dem XenDesktop-Installationsmedium im Ordner "Support/Tools/Scripts") oder mit dem Überwachungstool für PvD-Imageupdates (im Ordner "Support/Tools/PvdTool") können Sie die PvD-Datenträger in einem Katalog identifizieren, für die ein Update geplant ist und die fast voll sind.

Das Vorhandensein von Antivirenprodukten kann sich darauf auswirken, wie lange das Durchführen einer Bestandsaktualisierung oder eines Updates dauert. Die Leistung kann verbessert werden, wenn Sie CtxPvD.exe und CtxPvDSvc.exe der Ausschlussliste des Antivirenprodukts hinzufügen. Diese Dateien befinden sich im Ordner C:\Programme\Citrix\personal vdisk\bin. Durch das Ausschließen dieser ausführbaren Dateien vom Antivirenschscan kann die Bestandsaktualisierungs- und Imageupdateleistung um das Zehnfache beschleunigt werden.

- **Mehraufwand für unerwartete Zunahme (unerwartete Anwendungsinstallationen usw.)**

Kalkulieren Sie zusätzlichen Speicherplatz ein (entweder eine feste Menge oder einen Prozentsatz der vDisk-Größe), um auf unerwartete Anwendungsinstallationen von Benutzern während der Bereitstellung vorbereitet zu sein.

Konfigurieren von Größe und Zuordnung der persönlichen vDisk

Sie können den Algorithmus für die automatische Größenänderung, der die Größe der virtuellen Festplatte relativ zum Laufwerk P: festlegt, manuell anpassen, indem Sie die Ausgangsgröße der virtuellen Festplatte festlegen. Dies ist nützlich, wenn Sie beispielsweise wissen, dass Benutzer zahlreiche Anwendungen installieren werden, die nicht alle auf die virtuelle Festplatte passen werden, selbst wenn die Größe mit dem Algorithmus angepasst wurde. In dieser Situation können Sie die Ausgangsgröße des Anwendungsspeicherplatzes erhöhen, damit die vom Benutzer installierten Anwendungen genug Platz haben.

Passen Sie die Ausgangsgröße der virtuellen Festplatte am besten auf einem Masterimage an. Sie können die Größe der virtuellen Festplatte auch auf einem virtuellen Desktop anpassen, wenn ein Benutzer nicht genügend Speicherplatz für die Installation einer Anwendung hat. Sie müssen diesen Vorgang jedoch auf allen betroffenen virtuellen Desktops wiederholen, da Sie die Ausgangsgröße der virtuellen Festplatte nicht in einem bereits erstellten Katalog ändern können.

Stellen Sie sicher, dass die virtuelle Festplatte groß genug für das Speichern von Definitionsdateien von Antivirensoftware ist, da diese Dateien normalerweise sehr umfangreich sind.

Legen Sie die folgenden Registrierungsschlüssel in HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\personal vDisk\Config fest. (Ändern Sie keine anderen Einstellungen in diesem Registrierungsschlüssel.) Alle Einstellungen müssen auf dem Masterimage angegeben werden (außer für MinimumVHDSIZEinMB, was auf einer individuellen Maschine geändert werden kann). Auf dem Masterimage angegebene Einstellungen werden bei der nächsten Aktualisierung des Images angewendet.

- **MinimumVHDSIZEinMB**

Gibt die Mindestgröße (in Megabyte) des Anwendungsteils (C:) der persönlichen vDisk an. Die neue Größe muss größer als die vorhandene Größe und kleiner als die Größe des Datenträgers abzüglich PvDReservedSpaceMB sein.

Durch das Erhöhen dieses Werts wird freier Speicherplatz vom Profiteil der vDisk dem Laufwerk C: zugeteilt. Diese Einstellung wird ignoriert, wenn ein geringerer Wert als die aktuelle Größe des Laufwerks C: verwendet wird oder wenn EnableDynamicResizeOfAppContainer auf 0 eingestellt ist.

Standard = 2048

- **EnableDynamicResizeOfAppContainer**

Aktiviert oder deaktiviert den Algorithmus zur dynamischen Größenänderung.

- Bei der Einstellung auf 1 wird der Anwendungsspeicherplatz auf C: automatisch geändert, wenn der freie Speicherplatz auf C: unter 10 % fällt. Zulässige Werte sind 1 und 0. Ein Neustart muss durchgeführt werden, damit die Änderungen wirksam werden.
- Bei der Einstellung auf 0 wird die Größe der virtuellen Festplatte basierend auf der Methode ermittelt, die in XenDesktop-Versionen vor 7.x verwendet wurde.

Standard = 1

- **EnableUserProfileRedirection**

Aktiviert oder deaktiviert die Umleitung des Benutzerprofils auf die vDisk.

- Bei der Einstellung auf 1 leitet PvD das Benutzerprofil auf die vDisk um (standardmäßig auf P:). Profile werden entsprechend einem Windows-Standardprofil im Allgemeinen nach P:\Users umgeleitet. Durch die Umleitung werden die Profile beibehalten, falls der PvD-Desktop zurückgesetzt werden muss.
- Bei der Einstellung auf 0 ist der gesamte Speicherplatz auf der vDisk, abzüglich PvDReservedSpaceMB, dem Laufwerk C: (dem Anwendungsteil der vDisk) zugewiesen, und das vDisk-Laufwerk (P:) ist im Windows Explorer ausgeblendet. Citrix empfiehlt das Deaktivieren der Umleitung durch die Einstellung auf 0, wenn Sie Citrix Profilverwaltung oder eine andere Roamingprofillosung verwenden.

Durch diese Einstellung bleiben die Profile in C:\Users erhalten statt auf die vDisk umgeleitet zu werden, und sie ermöglicht der Roamingprofillosung das Verwalten der Profile.

Dieser Wert stellt sicher, dass der ganze Speicherplatz auf P: Anwendungen zugewiesen ist.

Bei einer Einstellung dieses Werts auf 0 wird davon ausgegangen, dass eine Profilverwaltungslösung eingesetzt wird. Das Deaktivieren der Profulumleitung ohne eine Roamingprofillosung einzusetzen wird nicht empfohlen, da die Profile sonst bei nachfolgenden PvD-Zurücksetzungsvorgängen gelöscht werden.

Ändern Sie diese Einstellung nicht, wenn das Image aktualisiert wird, da sie zwar nicht den Speicherort vorhandener Profile ändert, jedoch den gesamten Speicherplatz auf der PvD dem Laufwerk C: zuweist und die PvD ausblendet.

Konfigurieren Sie diesen Wert vor dem Bereitstellen eines Katalogs. Nach dem Bereitstellen eines Katalogs können Sie diesen Wert nicht mehr ändern.

Wichtig: Ab XenDesktop 7.1 werden Änderungen an diesem Wert nicht angewendet, wenn Sie ein Imageupdate ausführen. Legen Sie den Wert des Schlüssels fest, wenn Sie die Kataloge erstellen, von denen die Profile erstellt werden. Sie können das Umleitungsverhalten später nicht mehr ändern.

Standard = 1

- **PercentOfPvDForApps**

Legt die Teilung zwischen dem Anwendungsteil (C:) und dem Profiltteil der vDisk fest. Dieser Wert wird beim Erstellen neuer VMs verwendet sowie während Imageupdates, wenn EnableDynamicResizeOfAppContainer auf 0 eingestellt ist.

Das Ändern der Einstellung PercentOfPvDForApps macht nur einen Unterschied, wenn die Einstellung EnableDynamicResizeOfAppContainer auf 0 festgelegt ist. Standardmäßig ist EnableDynamicResizeOfAppContainer auf 1 (aktiviert) festgelegt, d. h. dass der AppContainer (Laufwerk C:) nur erweitert wird, wenn er fast voll ist (d. h. dynamisch), und zwar wenn weniger als 10 % freier Speicherplatz vorhanden ist.

Durch das Erhöhen von PercentOfPvDForApps wird nur der Gesamtspeicherplatz für die Apps erhöht. Der Speicherplatz wird jedoch nicht sofort verfügbar gemacht. Sie müssen zudem die Zuordnungsteilung im Masterimage konfigurieren, die dann beim nächsten Imageupdate angewendet wird.

Wenn Sie bereits einen Katalog mit Maschinen mit der Einstellung 1 für EnableDynamicResizeOfAppContainer generiert haben, ändern Sie die Einstellung im Masterimage für das nächste Update auf 0 und konfigurieren Sie die entsprechende Zuordnungsteilung. Die angeforderte Teilungsgröße wird eingehalten, solange sie größer als die aktuell zugeteilte Größe des Laufwerks C ist.

Wenn Sie komplette Kontrolle über die Speicherplatzteilung haben möchten, legen Sie den Wert auf 0 fest. Damit haben Sie die komplette Kontrolle über die Größe des Laufwerks C und die Erweiterung des Laufwerks ist nicht davon abhängig, dass ein Benutzer eine bestimmte Menge Speicherplatz unter dem Schwellenwert in Anspruch nimmt.

Standard = 50 % (beiden Teilen wird der gleiche Speicherplatz zugeteilt)

- **PvDReservedSpaceMB**

Legt die Größe des reservierten Speicherplatzes (in MB) auf der vDisk für das Speichern von Personal vDisk-Protokollen und anderen Daten fest.

Wenn in Ihrer Bereitstellung XenApp 6.5 (oder eine frühere Version) und Anwendungsstreaming verwendet wird, erhöhen Sie diesen Wert um die Größe des RadeCache.

Standard = 512

- **PvDResetUserGroup**

Gilt nur für XenDesktop 5.6: Ermöglicht der angegebenen Gruppe von Benutzern, eine persönliche vDisk zurücksetzen. Höhere XenDesktop-Releases verwenden dafür die delegierte Administration.

Weitere Einstellungen:

- **Windows Update Service:** Stellen Sie sicher, dass im Masterimage für Windows Updates die Einstellung Never Check for Update und für den Windows Update Service die Einstellung Disabled ausgewählt ist. Falls der Windows Update Service auf der PvD ausgeführt werden muss, wird durch die Einstellung Never Check for Update verhindert, dass die Updates auf den zugeordneten Maschinen installiert werden.
Dieser Dienst wird von Windows 8 Store benötigt, um Modern UI-Apps installieren zu können.
- **Windows Updates:** Diese schließen Updates für Internet Explorer ein und müssen auf das Masterimage angewendet werden.
- **Updates requiring restarts:** Bei auf das Masterimage angewendeten Windows Updates sind, abhängig von den im Update enthaltenen Patches, möglicherweise mehrere Neustarts erforderlich, damit die Installation vollständig ausgeführt wird. Stellen Sie vor der PvD-Bestandsaktualisierung sicher, dass Sie das Masterimage richtig neu starten,

damit die Installation von ausgeführten Windows Updates vollständig abgeschlossen wird.

- **Application updates:** Aktualisieren Sie Anwendungen, die auf dem Masterimage installiert sind, um Speicherplatz auf den vDisks von Benutzern zu sparen. Auf diese Weise vermeiden Sie den doppelten Aufwand, der durch das Durchführen von Updates auf den vDisks einzelner Benutzer entstehen würde.

Überlegungen zu Anwendungen auf dem Masterimage

Es kann zu Konflikten zwischen Software und der von PvD erstellten Benutzerumgebung kommen. Zum Vermeiden von Konflikten müssen Sie die Software auf dem Masterimage (nicht auf den individuellen Maschinen) installieren. Auch wenn es keine Konflikte zwischen der Software und der Ausführung von PvD gibt, empfiehlt Citrix, sie auf dem Masterimage zu installieren.

Folgende Anwendungen müssen auf dem Masterimage installiert werden:

- Agents und Clients (z. B. System Center Configuration Manager-Agent, App-V-Client, Citrix Receiver)
- Anwendungen, die vorrangige Starttreiber installieren oder ändern
- Anwendungen, die Drucker- oder Scannersoftware bzw. -treiber installieren
- Anwendungen, die den Windows-Netzwerkstapel ändern
- VM-Tools wie VMware Tools und XenServer Tools

Folgende Anwendungen sollten auf dem Masterimage installiert werden:

- Anwendungen, die einer großen Anzahl an Benutzern zur Verfügung gestellt werden. Deaktivieren Sie für die folgenden Fälle vor der Bereitstellung die Anwendungsupdates:
 - Unternehmensanwendungen, die Volumenlizenzierung verwenden, z. B. Microsoft Office und Microsoft SQL Server
 - Gängige Anwendungen wie Adobe Reader, Firefox und Chrome
- Große Anwendungen, z. B. SQL Server, Visual Studio und Anwendungsframeworks wie .NET

Die folgenden Empfehlungen und Einschränkungen gelten für Anwendungen, die Benutzer auf Maschinen mit persönlichen vDisks installiert haben. Einige dieser Empfehlungen können nicht durchgesetzt werden, wenn Benutzer Administratorrechte haben:

- Benutzer sollten Anwendungen nicht vom Masterimage deinstallieren und dann auf ihrer persönlichen vDisk neu installieren.
- Vorsicht beim Aktualisieren oder Deinstallieren von Anwendungen auf dem Masterimage. Nachdem Sie die Version einer Anwendung auf dem Image installiert haben, installiert ein Benutzer möglicherweise eine Add-On-Anwendung (z. B. ein Plug-In), das diese Version erfordert. Im Fall einer solchen Abhängigkeit kann es nach dem Aktualisieren oder Deinstallieren der Anwendung auf dem Image beim Add-On zu einer Fehlfunktion kommen. Beispiel: Microsoft Office 2010 ist auf einem Masterimage installiert und ein Benutzer installiert Visio 2010 auf der persönlichen vDisk. Bei einem Upgrade von Office auf dem Masterimage kann die lokal installierte Visio-Anwendung unbrauchbar werden.
- Software mit Lizenzen, die von Hardware abhängig sind (entweder durch ein Dongle oder signaturbasierte Hardware), wird nicht unterstützt.

Überlegungen zu Provisioning Services

Wenn Sie Provisioning Services mit PvD verwenden:

- Das SOAP-Dienstkonto muss zum Administratorknoten von Studio hinzugefügt werden und es muss über die Rolle "Maschinenadministrator" oder eine höhere Rolle verfügen. Dadurch wird sichergestellt, dass die PvD-Desktops in den Status "Preparing" versetzt werden, wenn die Provisioning Services (PVS) vDisk auf "Production" hochgestuft wird.
- Das Versionsverwaltungsfeature von Provisioning Services muss zum Aktualisieren der persönlichen vDisk verwendet werden. Wenn die Version auf "Production" hochgestuft wird, versetzt der SOAP-Dienst die PvD-Desktops in den Status "Preparing".

- Die Größe der persönlichen vDisk sollte immer größer als der Provisioning Services-Schreibcachedatenträger sein, da Provisioning Services sonst fälschlicherweise die persönliche vDisk als Schreibcache nehmen könnte.
- Nach dem Erstellen einer Bereitstellungsgruppe können Sie die persönliche vDisk mit dem [Überwachungstool für PvD-Imageupdates](#) oder den [Größenänderungs- und poolstats-Skripts](#) (personal-vdisk-poolstats.ps1) überwachen.

Veranschlagen Sie genug Schreibcache. Bei normalem Betrieb werden die meisten Benutzerschreibvorgänge (Änderungen) von PvD erfasst und an die persönliche vDisk umgeleitet. Daher könnten Sie theoretisch die Größe des Provisioning Services-Schreibcache verringern. Wenn PvD jedoch nicht aktiv ist, z. B. während Imageupdatevorgängen, kann ein kleines Provisioning Services-Schreibcache schnell voll sein und den Absturz von Maschinen verursachen.

Citrix empfiehlt, dass Sie die Größe des Provisioning Services-Schreibcache entsprechend den Empfehlungen für Provisioning Services veranschlagen und dann Speicherplatz hinzufügen, der dem Doppelten der virtuellen Festplattenvorlage auf dem Masterimage entspricht (für Zusammenführungen). Es ist unwahrscheinlich, dass ein Zusammenführungsvorgang so viel Speicherplatz beansprucht, aber es ist möglich.

Wenn Sie mit Provisioning Services einen Katalog mit PvD-aktivierten Maschinen bereitstellen:

- Folgen Sie den Anweisungen in der Provisioning Services-Dokumentation.
- Sie können die Drosselungseinstellungen für Energieaktionen ändern, indem Sie die Verbindung in Studio bearbeiten.
- Zum Aktualisieren der Provisioning Services-vDisk installieren bzw. aktualisieren Sie Anwendungen und Software und starten Sie die vDisk neu. Aktualisieren Sie dann den PvD-Bestand und fahren Sie die VM herunter. Stufen Sie anschließend die neue Version auf "Production" hoch. Die PvD-Desktops im Katalog sollten automatisch in den Zustand "Preparing" versetzt werden. Wenn dies nicht der Fall ist, prüfen Sie, ob das SOAP-Dienstkonto Maschinenadministrator- oder höhere Privilegien auf dem Controller hat.

Mit dem Testmodusfeature von Provisioning Services können Sie einen Testkatalog mit Maschinen erstellen, die ein aktualisiertes Masterimage verwenden. Wenn die Tests die Funktionsfähigkeit des Katalogs bestätigen, stufen Sie ihn auf "Production" hoch.

Überlegungen zu den Maschinenerstellungsdiensten

Wenn Sie mit den Maschinenerstellungsdiensten (MCS) einen Katalog mit PvD-aktivierten Maschinen bereitstellen:

- Folgen Sie den Anweisungen in der XenDesktop-Dokumentation.
- Aktualisieren Sie nach dem Erstellen des Masterimages den PvD-Bestand und schalten Sie dann die VM aus (PvD funktioniert nicht ordnungsgemäß, wenn Sie die VM nicht ausschalten). Erstellen Sie einen Snapshot vom Masterimage.
- Geben Sie im Assistenten zum Erstellen von Maschinenkatalogen die Größe und den Laufwerksbuchstaben der persönlichen vDisk an.
- Nach dem Erstellen einer Bereitstellungsgruppe können Sie die persönliche vDisk mit dem [Überwachungstool für PvD-Imageupdates](#) oder den [Größenänderungs- und poolstats-Skripts](#) (personal-vdisk-poolstats.ps1) überwachen.
- Sie können die Drosselungseinstellungen für Energieaktionen ändern, indem Sie die Verbindung in Studio bearbeiten.
- Wenn Sie das Masterimage aktualisieren, nehmen Sie nach dem Aktualisieren der Anwendungen und der anderen Software auf dem Image den PvD-Bestand auf und schalten Sie anschließend die VM aus. Erstellen Sie einen Snapshot vom Masterimage.
- Überprüfen Sie mit dem Überwachungstool für PvD-Imageupdates oder mit dem Skript "personal-vdisk-poolstats.ps1", ob genügend Speicherplatz auf jeder PvD-aktivierten VM vorhanden ist, die das aktualisierte Masterimage verwendet.
- Nach dem Aktualisieren des Maschinenkatalogs werden die PvD-Desktops in den Zustand "Preparing" versetzt, während sie die Änderungen am neuen Masterimage verarbeiten. Die Desktops werden entsprechend der während des Maschinenupdates festgelegten Rolloutstrategie aktualisiert.
- Während die PvD im Zustand "Preparing" ist, überwachen Sie sie mit dem Überwachungstool für PvD-Imageupdates oder mit dem Skript "personal-vdisk-poolstats.ps1".

Ausschließen von Dateien und Ordner von vDisks

Mit Regeldateien schließen Sie Dateien und Ordner von vDisks aus. Dies ist möglich, während die persönlichen vDisks bereitgestellt werden. Die Regeldateien werden `custom_*_rules.template.txt` genannt und befinden sich im Ordner `\config`. Anmerkungen in den einzelnen Dateien erhalten zusätzliche Informationen.

Durchführen einer Bestandsaktualisierung beim Aktualisieren eines Masterimages

Wenn Sie PvD nach einer Aktualisierung des Masterimages aktivieren, muss der Bestand des Datenträgers aktualisiert und ein neuer Snapshot erstellt werden.

Wenn Sie eine Anwendung installieren, die Binärdateien im Benutzerprofil des Administrators ablegt, ist die Anwendung nicht für Benutzer freigegebener virtueller Desktops (einschließlich solcher, die auf gepoolten Maschinenkatalogen basieren und mit PvD-Maschinenkatalogen gepoolt sind) verfügbar, da Masterimages nicht von Benutzern sondern von Administratoren verwaltet werden. Benutzer müssen solche Anwendungen selber installieren.

Es ist ratsam, nach jedem Schritt in diesem Verfahren einen Snapshot des Images zu erstellen:

1. Aktualisieren Sie das Masterimage, indem Sie auf der Maschine Anwendungen oder Betriebssystemupdates installieren und das System konfigurieren.
Bei Masterimages, die auf Windows XP basieren und die Sie mit persönlichen vDisks bereitstellen möchten, müssen Sie sicherstellen, dass keine Dialogfelder offen sind (z. B. Meldungen zur Bestätigung von Softwareinstallationen oder Aufforderungen, nicht signierte Treiber zu verwenden). Offene Dialogfelder auf Masterimages in dieser Umgebung verhindern, dass sich der VDA beim Delivery Controller registriert. Sie können Aufforderungen für nicht signierte Treiber in der Systemsteuerung verhindern. Navigieren Sie zu "System > Hardware > Treibersignierung" und wählen Sie die Option zum Ignorieren von Warnungen.
2. Fahren Sie die Maschine herunter. Klicken Sie bei Windows 7-Maschinen auf Abbrechen, wenn Citrix Personal vDisk das Herunterfahren behindert.
3. Klicken Sie im Citrix Personal vDisk-Dialogfeld auf Bestand aktualisieren. Dieser Vorgang kann einige Minuten dauern.
Wichtig: Wenn Sie das anschließende Herunterfahren unterbrechen (selbst wenn Sie nur eine kleine Änderung am Image vornehmen), stimmt der Bestand der persönlichen vDisk nicht mehr mit dem Masterimage überein. Dies führt dazu, dass das Personal vDisk-Feature nicht mehr funktioniert. Wenn Sie das Herunterfahren unterbrechen, müssen Sie die Maschine neu starten, herunterfahren und auf "Bestand aktualisieren" klicken, wenn Sie dazu aufgefordert werden.
4. Erstellen Sie, nachdem der Bestandvorgang die Maschine heruntergefahren hat, einen Snapshot des Masterimages.

Sie können einen Bestand in eine Netzwerkfreigabe exportieren und ihn anschließend in ein Masterimage importieren. Weitere Informationen finden Sie unter [Exportieren und Importieren eines PvD-Bestands](#).

Konfigurieren von Drosselungseinstellungen für Verbindungen

Der Citrix Brokerdienst steuert den Energiezustand der Maschinen, die Desktops und Anwendungen bereitstellen. Der Brokerdienst kann mehrere Hypervisoren über einen Delivery Controller steuern. Die Interaktion zwischen einem Controller und dem Hypervisor wird durch Broker-Energieaktionen gesteuert. Aktionen, die den Energiezustand einer Maschine ändern, wird eine Priorität zugewiesen und dann werden sie über einen Drosselungsmechanismus an den Hypervisor gesendet, damit es nicht zur Überlastung kommt. Die folgenden Einstellungen wirken sich auf die Drosselung aus. Diese Werte werden festgelegt, indem Sie eine Verbindung (Seite "Erweitert") in Studio bearbeiten.

Konfigurieren von Drosselungswerten für eine Verbindung

1. Wählen Sie im Studio-Navigationsbereich Konfiguration > Hosting.
2. Wählen Sie die Verbindung und dann im Aktionsbereich Verbindung bearbeiten.

3. Sie können die folgenden Werte ändern:

- **Gleichzeitige Aktionen (alle Typen):** Das zulässige Maximum für gleichzeitig ausgeführte Energieaktionen. Diese Einstellung wird als absoluter Wert und als Prozentsatz der Verbindung mit dem Hypervisor angegeben. Der niedrigere der beiden Werte wird verwendet.
Standard = 100 absolut, 20%
- **Gleichzeitige Updates für Personal vDisk-Bestand:** Das zulässige Maximum für gleichzeitige Personal vDisk-Energieaktionen. Diese Einstellung wird als absoluter Wert und als Prozentsatz der Verbindung angegeben. Der niedrigere der beiden Werte wird verwendet.
Standard = 50 absolut, 25%

Sie kalkulieren den absoluten Wert, indem Sie den Gesamtwert für IOPS (TIOPS) bestimmen, den der Datenträger des Endbenutzers unterstützt (dies sollte vom Hersteller festgelegt sein oder berechnet werden). Veranschlagen Sie 350 IOPS pro VM (IOPS/VM), um die Anzahl der VMs zu bestimmen, die jeweils auf dem Datenträger aktiv sein können. Sie berechnen diesen Wert, indem Sie den Gesamtwert für IOPS durch IOPS/VM teilen.

Beispiel: Wenn der Wert für den Datenträger des Endbenutzers 14.000 IOPS ist, ist die Anzahl der aktiven VMs $14.000 \text{ IOPS} : 350 \text{ IOPS/VM} = 40$.

- **Höchstanzahl neue Aktionen pro Minute:** Die maximale Anzahl der neuen Energieaktionen, die pro Minute an den Hypervisor gesendet werden können. Sie wird als absoluter Wert angegeben.
Standard = 10

Identifizieren der optimalen Werte für diese Einstellungen in der Bereitstellung:

1. Messen Sie mit den Standardwerten die Reaktionszeit für das Imageupdate eines Testkatalogs. Dies ist die Differenz zwischen der Startzeit eines Imageupdates (T1) und dem Zeitpunkt, wenn der VDA auf der letzten Maschine des Katalogs beim Controller registriert wird (T2). Reaktionszeit = $T2 - T1$.
2. Messen Sie die Ein- und Ausgabevorgänge pro Sekunde (IOPS) auf dem Hypervisorspeicher während des Imageupdates. Diese Daten können als Benchmark für die Optimierung dienen. (Die Standardwerte sind möglicherweise die beste Einstellung. Unter Umständen erreicht das System den maximalen IOPS-Wert, sodass die Einstellungswerte herabgesetzt werden müssen.)
3. Ändern Sie den Wert für "Gleichzeitige Updates für Personal vDisk-Bestand" wie unten beschrieben und lassen Sie alle anderen Einstellungen unverändert.
 1. Erhöhen Sie den Wert um 10 und messen Sie die Reaktionszeit nach jeder Änderung. Fahren Sie fort, den Wert um 10 zu erhöhen und messen Sie das Ergebnis, bis die Reaktionszeit abnimmt oder keine Änderung mehr auftritt.
 2. Wenn durch das Erhöhen des Werts im vorherigen Schritt keine Verbesserung erzielt wurde, verringern Sie den Wert schrittweise um 10 und messen Sie die Reaktionszeit nach jeder Verringerung. Wiederholen Sie diesen Vorgang, bis sich die Reaktionszeit nicht mehr ändert oder verbessert. Dieser Wert ist wahrscheinlich der optimale Wert für die PvD-Energieaktion.
4. Wenn Sie den Einstellungswert für die PvD-Energieaktion festgelegt haben, optimieren Sie nacheinander die Werte für die gleichzeitigen Aktionen (alle Typen) und die Höchstanzahl der neuen Aktionen pro Minute. Folgen Sie den oben erläuterten Schritten (schrittweises Erhöhen und Verringern der Werte), um verschiedene Werte zu testen.

Verwenden von Microsoft System Center Configuration Manager 2007 mit PvD

System Center Configuration Manager (Configuration Manager) 2012 erfordert keine besondere Konfiguration und kann auf die gleiche Weise wie alle anderen Anwendungen auf dem Masterimage installiert werden. Die folgenden Informationen gelten nur für System Center Configuration Manager 2007. Configuration Manager-Versionen vor Configuration Manager 2007 werden nicht unterstützt.

Führen Sie die folgenden Schritte aus, um die Configuration Manager 2007 Agent-Software in einer PvD-Umgebung zu verwenden.

1. Installieren Sie den Client-Agent auf dem Masterimage.
 1. Installieren Sie den Configuration Manager-Client auf dem Masterimage.
 2. Beenden Sie den ccmexec-Dienst (SMS-Agent) und deaktivieren Sie ihn.
 3. Löschen Sie SMS- oder Clientzertifikate wie folgt aus dem Zertifikatspeicher des lokalen Computers:
 - Gemischter Modus: Certificates (Lokaler Computer)\SMS\Certificates
 - Einheitlicher Modus
 - Certificates (Lokaler Computer)\Personal\Certificates
 - Löschen Sie das Clientzertifikat, das von Ihrer Zertifizierungsstelle ausgestellt wurde (normalerweise eine interne PKI).
 4. Löschen Sie C:\Windows\smscfg.ini oder benennen Sie die Datei um.
2. Entfernen Sie Informationen, die den Client eindeutig identifizieren.
 1. (Optional) Löschen Sie die Protokolldateien unter C:\Windows\System32\CCM\Logs oder verschieben Sie sie.
 2. Installieren Sie ggf. den Virtual Delivery Agent und nehmen Sie den PvD-Bestand auf.
 3. Fahren Sie das Masterimage herunter, erstellen Sie einen Snapshot und erstellen Sie dann mit diesem Snapshot einen Maschinenkatalog.
3. Überprüfen Sie Personal vDisk und starten Sie die Dienste. Führen Sie diese Schritte einmal auf jedem PvD-Desktop aus, nachdem er zum ersten Mal gestartet wurde. Dazu können Sie beispielsweise ein Domänen-GPO verwenden.
 - Bestätigen Sie, dass PvD aktiv ist, indem Sie prüfen, ob der Registrierungsschlüssel HKLM\Software\Citrix\personal vDisk\config\virtual vorhanden ist.
 - Stellen Sie den ccmexec-Dienst (SMS-Agent) auf "Automatic" ein und starten Sie den Dienst. Der Configuration Manager-Client kontaktiert den Configuration Manager-Server und ruft neue, eindeutige Zertifikate und GUIDs ab.

Tools

Sep 29, 2015

Mit den folgenden Tools und Hilfsprogrammen können Sie PVD-Vorgänge anpassen, vereinfachen und überwachen.

Benutzerdefinierte Regeldateien

Mit den von PVD bereitgestellten benutzerdefinierten Regeldateien können Sie das folgende Standardverhalten von PVD-Imageupdates ändern:

- Die Sichtbarkeit von Dateien auf der PVD
- Die Art der Zusammenführung von vorgenommenen Änderungen
- Einstellungen zur Beschreibbarkeit der Dateien

Detaillierte Anweisungen zu den benutzerdefinierten Regeldateien und dem CoW-Feature finden Sie in den Kommentaren zu den Dateien, die sich unter C:\ProgramData\Citrix\personal vDisk\Config auf der Maschine befinden, auf der PVD installiert ist. Die Dateien mit dem Namen custom_* erläutern die Regeln und wie sie aktiviert werden.

Ändern der Größe und poolstats-Skripts

Es gibt zwei Skripts zum Überwachen und Verwalten der Größe der PVDs. Sie befinden sich im Ordner "Support\Tools\Scripts" auf dem XenDesktop-Installationsmedium. Sie können auch das Überwachungstool für PVD-Imageupdates verwenden, das sich im Ordner "Support\Tools\Scripts\PvdTool" befindet. Weitere Informationen finden Sie unter <http://blogs.citrix.com/2014/06/02/introducing-the-pvd-image-update-monitoring-tool/>.

Verwenden Sie "resize-personalvdisk-pool.ps1" zum Vergrößern der PVDs in allen Desktops eines Katalogs. Die folgenden Snap-Ins oder Module für den Hypervisor müssen auf der Maschine installiert werden, auf der Studio ausgeführt wird:

- XenServer erfordert XenServerPSSnapin
- vCenter erfordert vSphere PowerCLI
- System Center Virtual Machine Manager erfordert die VMM-Konsole

Mit "personal-vdisk-poolstats.ps1" können Sie den Status von Imageupdates überprüfen sowie den Speicherplatz für Anwendungen und Benutzerprofile in einer PVD-Gruppe. Führen Sie das Skript vor dem Update eines Images aus, um zu prüfen, ob Desktops genug Speicherplatz haben. Dadurch werden Fehler während des Updates verhindert. Für das Skript muss die Windows Management Instrumentation (WMI-In)-Firewall auf den PVD-Desktops aktiviert sein. Sie können die Firewall auf dem Masterimage aktivieren oder über GPO.

Wenn ein Imageupdate fehlschlägt, zeigt der Eintrag in der Spalte "Update" die Ursache an.

Zurücksetzen des Anwendungsbereichs

Wenn ein Desktop durch die Installation einer fehlerhaften Anwendung oder aus einem anderen Grund beschädigt wird, können Sie den Anwendungsbereich der PVD auf den (leeren) Herstellerstandard zurücksetzen. Beim Zurücksetzen bleiben die Benutzerprofilaten erhalten.

So setzen Sie den Anwendungsbereich der PVD zurück:

- Melden Sie sich am Desktop des Benutzers als Administrator an. Starten Sie eine Eingabeaufforderung und führen Sie den Befehl C:\Programme\Citrix\Personal vDisk\bin\CtxPvD.exe -s Reset aus.
- Navigieren Sie in Citrix Director zum Desktop des Benutzers. Klicken Sie auf Reset Personal vDisk und anschließend auf OK.

Exportieren und Importieren eines PVD-Bestands

Der Imageupdateprozess ist ein zentraler Teil der Bereitstellung neuer Images auf PVD-Desktops und umfasst Anpassungen, damit vorhandene persönliche vDisks mit dem neuen Basisimage funktionieren. Bei Bereitstellungen, die Maschinenerstellungsdienste (MCS) verwenden, können Sie einen Bestand von einer aktiven VM auf eine Netzwerkfreigabe exportieren und dann in ein Masterimage importieren. Mit diesem Bestand auf dem Masterimage wird eine Differenz berechnet. Obwohl das Feature zum Exportieren bzw. Importieren des Bestands nicht verwendet werden muss, kann es die Leistung des Imageupdateprozesses verbessern.

Sie müssen ein Administrator sein, um das Feature zum Exportieren/Importieren des Bestands zu verwenden. Authentifizieren Sie sich für den Export/Import ggf. bei der Dateifreigabe mit "net user". Der Benutzerkontext muss auf alle für den Export/Import verwendeten Dateifreigaben zugreifen können.

- Führen Sie zum Exportieren eines Bestands den Exportbefehl als Administrator auf einer Maschine aus, auf der sich ein VDA mit aktivierter PVD befindet (Mindestversion 7.6):
Ctxpvdsvc.exe exportinventory "<Pfad zum Exportspeicherort>"

Die Software erkennt den Speicherort des aktuellen Bestands und exportiert den Bestand an den angegebenen Speicherort in einen Ordner mit dem Namen "ExportedPvdInventory". Hier ein Auszug aus der Befehlsausgabe:

- C:\Program Files\Citrix\personal vDisk\bin> .\CtxPvDsvc.exe exportinventory \\share location\ExportedInventory Current inventory source location C:\CitrixPvD\Settings\Inventory\VER-LAS ... Exporting
- Zum Importieren eines zuvor exportierten Bestands führen Sie den Importbefehl als Administrator auf dem Masterimage aus:

Importieren

Führen Sie den Importbefehl als Administrator auf dem Masterimage aus.

Ctxpvdsvc.exe importinventory "<Pfad zum exportierten Bestand>"

Der <Pfad zum exportierten Bestand> muss der vollständige Pfad für die Bestandsdateien sein; in der Regel ist das <Netzwerkspeicherort\ExportedPvdInventory>.

Der Bestand wird aus dem Importspeicherort abgerufen (zuvor wurde er mit dem Befehl zum Exportieren des Bestands hierher exportiert) und in den Bestandspeicher auf dem Masterimage importiert. Hier ein Auszug aus der Befehlsausgabe:

C:\Program Files\Citrix\personal vDisk\bin> .\CtxPvDsvc.exe importinventory \\share location\ExportedInventory\ExportedPvdInventory Importing inventory \\share location\ExportedInventory\ExportedPvd
Nach dem Exportvorgang sollte die Netzwerkfreigabe die folgenden Dateinamen enthalten. Nach dem Importvorgang sollte der Bestandsspeicher auf dem Masterimage die gleichen Dateinamen enthalten.

- 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

Sichern und Wiederherstellen

Wichtig: Die Skripts verschieben die PVDs nicht an einen neuen Speicherort. Diesen Vorgang müssen Sie auf eine andere Weise ausführen.

Zwei PowerShell-Skripts im Ordner "Support\Tools\Scripts" auf dem Installationsmedium des Produkts ermöglichen das Sichern und Wiederherstellen persönlicher vDisks. Mit den Skripten zum Sichern und Wiederherstellen können Sie vorhandene PVDs und Benutzerzuweisungen von einem Katalog zu einem anderen migrieren. Dies kann nützlich sein, wenn Sie den PVD-Speicher ändern müssen. Das Skript zum Sichern erstellt eine XML-Datei mit Metadaten aus einem bestehenden Katalog. Die Metadaten enthalten den aktuellen Pfad der PVDs im Speicher und die Benutzerzuweisungen zu den PVDs. Das Skript zum Wiederherstellen verwendet die XML-Datei, um die PVDs einem neuen Katalog zuzuweisen und ihnen die richtigen Benutzer zuzuordnen.

- migration-backup.ps1 zeichnet die Zuordnung zwischen jedem Benutzer und dessen persönlicher vDisk in einem Maschinenkatalog auf und speichert diese Informationen in einer XML-Datei.
- migration-restore.ps1 erstellt mit der XML-Datei den Desktop des Benutzers in einem Maschinenkatalog neu.

Vor dem Sichern und Wiederherstellen beachten Sie Folgendes:

- Die Skripts verwenden die Hypervisor-API. Daher muss das PowerShell-Snap-In des Hypervisors auf dem Controller, auf dem die Skripts ausgeführt werden, installiert sein.
- Führen Sie die Skripts von einem Speicherort aus, der Zugriff auf den Controller hat, auf dem der Maschinenkatalog erstellt wurde.
- Die Skripts werden auf folgenden Hypervisor-Plattformen unterstützt: Citrix XenServer, Microsoft Hyper-V und VMware ESX.

Sichern eines Maschinenkatalogs

Führen Sie ein Backup durch, wenn ein Maschinenkatalog geändert wird. Sie können ein Backup durchführen, während die Maschinen im Katalog aktiv sind.

Verwenden Sie migration-backup.ps1 zum Sichern beliebiger Maschinenkataloge, die persönliche vDisks enthalten. Das Skript fordert die Angabe des Namens des Maschinenkatalogs und der Verbindungsinformationen für den Hypervisor an. Anschließend durchläuft es sämtliche Benutzern zugewiesene Maschinen im Maschinenkatalog und speichert für jede Maschine die Zuordnung zwischen persönlichem vDisk-Speicher und dem jeweiligen Benutzer. Diese Informationen werden in einer XML-Datei gespeichert, die folgende Struktur hat:

- PvdMigration.hypervisor.Type unterstützt VMware ESX, Citrix XenServer und Microsoft Hyper-V.
- PvdMigration.PVD speichert Informationen dazu, wo die persönliche vDisk gespeichert und welchem Benutzer sie zugeordnet ist.
- PvdMigration.PVD.DiskId ist der eindeutige Bezeichner der vDisk auf dem Hypervisor, auf dem das Backup erstellt wurde.
- PvdMigration.PVD.DiskName ist der Name der VHD- bzw. VMDK-Datei.
- PvdMigration.PVD.SRName ist der Name des Speicheranbieters, wenn das Backup erstellt wurde.
- PvdMigration.PVD.SRID ist der eindeutige Bezeichner des Speicheranbieters auf dem Hypervisor, auf dem das Backup erstellt wurde.
- PvdMigration.PVD.UserName ist der Name des Benutzers, der der vDisk zugeordnet ist.
- PvdMigration.PVD.UserId ist die SID des Benutzers, der der vDisk zugeordnet ist.
- PvdMigration.PVD.State ist der Status der vDisk. Der Status kann "backed up" oder "processed" lauten. Er lautet "backed up" nach dem ersten Backup und ändert sich in "processed", wenn die XML-Datei für eine Wiederherstellung vom Backup verwendet wurde.

Wiederherstellen eines Maschinenkatalogs

Beachten Sie vor dem Wiederherstellen Folgendes:

- Sie können nur einen Maschinenkatalog wiederherstellen, der dasselbe Masterimage hat wie der gesicherte Maschinenkatalog.
- Sie müssen ein neues Masterimage erstellen, indem Sie den Bestand des Masterimages aktualisieren, auf dessen Basis der gesicherte Maschinenkatalog erstellt wurde.

Stellen Sie mit migration-restore.ps1 beliebige Maschinenkataloge mit persönlichen vDisks wieder her. Das Skript erfordert folgende Eingaben:

- Während des Backups erstellte XML-Datei.
- Name des Maschinenkatalogs, der wiederhergestellt wird.
- Name des Speicherorts an dem die nicht zugewiesenen persönlichen vDisks gespeichert sind. Dies ist in der XML-Datei aufgelistet.
- Hypervisor-Verbindungsinformationen.

Das Skript migration-restore.ps1 findet alle nicht zugewiesenen Maschinen im Maschinenkatalog und weist ihnen Benutzer zu. Außerdem fügt es die persönlichen vDisks der Benutzer den Maschinen an.

Beispielszenario 1: Wiederherstellung eines Maschinenkatalogs und seiner persönlichen vDisks mit neuen Maschinenennamen

In diesem Szenario werden ein ganzer Maschinenkatalog und die persönlichen vDisks, die den enthaltenen Maschinen angefügt sind, wiederhergestellt. Die Maschinen erhalten neue Namen. Dies kann auftreten, wenn der Hypervisor oder ein Speicherhost ausgefallen ist oder wenn Benutzer in eine neue Infrastruktur migriert werden.

1. Führen Sie migration-backup.ps1 aus, um die Zuweisung zwischen Benutzern und persönlichen vDisks in der XML-Datei aufzuzeichnen.
2. Verschieben oder sammeln Sie mit einer Backuplösung die persönlichen vDisks aus dem ursprünglichen Maschinenkatalog auf einen Datenträger:
 - VMware ESX oder Microsoft Hyper-V: Persönliche vDisks sind in dem vom Controller festgelegten Speicher in einem Ordner, der den Namen der Maschine enthält, dem die vDisk angefügt ist.
 - Citrix XenServer: Persönliche vDisks sind im Stamm des vom Controller festgelegten Speichers. Der Name jeder vDisk ist eine GUID.
3. Stellen Sie die persönlichen vDisks aus dem ursprünglichen Maschinenkatalog mit einer Speicherbackuplösung wieder her:
 - ESX oder Hyper-V: Platzieren Sie die vDisks in einen neuen Ordner auf der neuen Speicherressource. Alternativ können Sie die vDisks im ursprünglichen Pfad auf der neuen Speicherressource lassen.
 - XenServer: Platzieren Sie die vDisks im Stamm der neuen Speicherressource.
4. Erstellen Sie einen Provisioning Services-vDisk- oder Maschinenerstellungsdienste-Snapshot des Masterimages, mit dem Sie den ausgefallenen Maschinenkatalog erstellt haben.
5. Führen Sie Bestand aktualisieren über das Startmenü auf der vDisk oder dem Snapshot aus.
6. Erstellen Sie den Maschinenkatalog in Studio neu und verwenden Sie dabei eine andere Benennungskonvention als für den ausgefallenen ursprünglichen Maschinenkatalog. Dadurch wird ein Katalog neuer Maschinen mit jeweils einer neuen persönlichen vDisk erstellt, die die Sitedatenbank erkennt.
7. Stellen Sie sicher, dass der neu erstellte Maschinenkatalog der richtigen Bereitstellungsgruppe zugewiesen ist.
8. Stellen Sie sicher, dass die Bereitstellungsgruppe im Wartungsmodus ist und die enthaltenen Maschinen heruntergefahren sind.
9. Bearbeiten Sie die vom Backupskript generierte XML-Datei:
 - ESX oder Hyper-V: Wenn Sie vDisks in Schritt 3 in einem neuen Ordner auf der neuen Speicherressource wiederhergestellt haben, ersetzen Sie für jeden PVD-Abschnitt in der Datei den Ordernamen in DiskName durch den Speicherort der wiederhergestellten vDisks. Wenn Sie die vDisks im ursprünglichen Pfad im neuen Speicher wiederhergestellt haben, überspringen Sie diesen Schritt.
 - XenServer: Überspringen Sie diesen Schritt.
10. Führen Sie auf dem Controller migration-restore.ps1 aus und geben Sie den Namen der XML-Datei und den Speicherort der gesicherten vDisks an.

Beispielszenario 2: Wiederherstellung eines Maschinenkatalogs und seiner persönlichen vDisks unter Wiederverwenden der bestehenden Maschinenennamen

In diesem Szenario werden ein ganzer Maschinenkatalog und die persönlichen vDisks, die den enthaltenen Maschinen angefügt sind, wiederhergestellt. Die bestehenden Namen der (ausgefallenen) Maschinen werden wiederverwendet. Dieses Szenario ist möglich, wenn der Hypervisor oder ein Speicherhost ausgefallen ist.

1. Führen Sie migration-backup.ps1 aus, um die Zuordnung zwischen Benutzern und persönlichen vDisks aufzuzeichnen.
2. Verschieben oder sammeln Sie mit einer Backuplösung die persönlichen vDisks aus dem ursprünglichen Maschinenkatalog auf einen Datenträger:
 - ESX oder Hyper-V: Persönliche vDisks sind in dem vom Controller festgelegten Speicher in einem Ordner, der den Namen der Maschine enthält, dem die vDisk angefügt ist.
 - XenServer: Persönliche vDisks sind im Stamm des vom Controller festgelegten Speichers. Der Name jeder vDisk ist eine GUID.
3. Stellen Sie die persönlichen vDisks aus dem ursprünglichen Maschinenkatalog mit einer Speicherbackuplösung wieder her:
 - ESX oder Hyper-V: Platzieren Sie die vDisks in einen neuen Ordner auf der neuen Speicherressource.
 - XenServer: Platzieren Sie die vDisks im Stamm der neuen Speicherressource.
4. Erstellen Sie einen Provisioning Services-vDisk- oder Maschinenerstellungsdienste-Snapshot des Masterimages, mit dem Sie den ausgefallenen Maschinenkatalog erstellt haben.
5. Führen Sie Bestand aktualisieren über das Startmenü auf der vDisk oder dem Snapshot aus.
6. Erstellen Sie den Maschinenkatalog in Studio neu und verwenden Sie dabei die gleiche Benennungskonvention wie für den ausgefallenen Maschinenkatalog. Dadurch wird ein Katalog neuer Maschinen mit jeweils einer neuen persönlichen vDisk erstellt, die die Sitedatenbank erkennt.
7. Stellen Sie sicher, dass der neu erstellte Maschinenkatalog der richtigen Bereitstellungsgruppe zugewiesen ist.
8. Stellen Sie sicher, dass die Desktopgruppe im Wartungsmodus ist und die enthaltenen Maschinen heruntergefahren sind.
9. Bearbeiten Sie die vom Backupskript generierte XML-Datei:
 - ESX oder Hyper-V: Ersetzen Sie für jeden PVD-Abschnitt in der Datei den Ordernamen in DiskName durch den Speicherort der wiederhergestellten vDisks.
 - XenServer: Überspringen Sie diesen Schritt.
10. Führen Sie das Skript migration-restore.ps1 auf dem Controller aus und verwenden Sie die geänderte XML-Datei als Eingabe. Das Skript fügt die vDisks an, ohne sie zu verschieben.
11. Überprüfen Sie, ob die Benutzerdaten erfolgreich wiederhergestellt wurden.

Beispielszenario 3: Wiederherstellung einer Teilmenge persönlicher vDisks in einem Maschinenkatalog

In diesem Szenario sind nur einige der persönlichen vDisks in einem Maschinenkatalog ausgefallen und werden wiederhergestellt. Die virtuellen Maschinen im Katalog sind nicht ausgefallen.

1. Führen Sie migration-backup.ps1 aus, um die Zuweisung zwischen Benutzern und persönlichen vDisks in der XML-Datei aufzuzeichnen.
2. Die XML-Datei enthält einen PVD-Abschnitt für jeden Benutzer im Maschinenkatalog. Entfernen Sie alle Benutzer, deren persönliche vDisks nicht wiederhergestellt werden müssen, und die zugehörigen Abschnitte aus der Datei.
3. Stellen Sie die persönlichen vDisks mit einer Backuplösung aus dem ursprünglichen Maschinenkatalog wie in einem der anderen Szenarios beschrieben wieder her:
 - Zum Verwenden neuer Maschinennamen folgen Sie Beispielszenario 1.
 - Zur Erhalten bestehenden Maschinennamen folgen Sie Beispielszenario 2.
4. Stellen Sie sicher, dass der Katalog genügend nicht zugewiesene Maschinen hat. Fügen Sie, falls erforderlich, Maschinen hinzu. Sie benötigen eine neue Maschine für jeden Benutzer, dessen vDisk Sie wiederherstellen möchten.
5. Stellen Sie sicher, dass die Desktopgruppe im Wartungsmodus ist und die enthaltenen Maschinen heruntergefahren sind.
6. Führen Sie auf dem Controller migration-restore.ps1 mit der geänderten XML-Datei als Eingabe aus.
7. Überprüfen Sie, ob die Benutzerdaten erfolgreich wiederhergestellt wurden.

Anzeigen, Meldungen und Problembehandlung

Sep 29, 2015

Wenn Sie in Studio eine PvD-aktivierte Maschine in einem Maschinenkatalog auswählen, können Sie auf der Registerkarte "PvD" den Überwachungsstatus während Imageupdates sowie die geschätzte Abschlusszeit und den Fortschritt verfolgen. Folgende Zustandsanzeigen sind während eines Imageupdates möglich: Ready, Preparing, Waiting, Failed und Requested.

Ein Imageupdate kann aus verschiedenen Gründen fehlschlagen, einschließlich zu wenig Speicherplatz oder weil ein Desktop die PvD nicht rechtzeitig findet. Wenn Studio angibt, dass ein Imageupdate fehlgeschlagen ist, wird ein Fehlercode mit beschreibendem Text angezeigt, um Sie bei der Problembehandlung zu unterstützen. Verwenden Sie das Überwachungstool für PvD-Imageupdates oder das Skript "personal-vdisk-poolstats.ps1" zum Überwachen des Imageupdatevorgangs und um Fehlercodes für das Problem zu erhalten.

Wenn ein Imageupdate fehlschlägt, finden Sie in den folgenden Protokolldateien weitere Informationen zur Problembehandlung:

- PvD-Dienstprotokoll: C:\ProgramData\Citrix\personal vDisk\Logs\PvDSvc.log.txt
- PvD-Aktivierungsprotokoll: P:\PVDLOGS\PvDActivation.log.txt

Der aktuelle Inhalt ist am Ende der Protokolldatei.

Fehlermeldungen: 7.6 und höher

Die folgenden Fehler gelten nur für PvD Version 7.6 und höher:

- **Ein interner Fehler ist aufgetreten. Weitere Informationen finden Sie in den Personal vDisk-Protokollen.**
Fehlercode% Disk (% s)

Dieser Code gilt für alle nicht kategorisierten Fehler und hat daher keinen numerischen Wert. Alle unerwarteten Fehler während der Bestandserstellung oder dem Update von Personal vDisk sind mit diesem Fehlercode gekennzeichnet.

- Sammeln Sie die Protokolle und wenden Sie sich an den Citrix Support.
- Wenn dieser Fehler während der Aktualisierung des Katalogs auftritt, führen Sie ein Rollback des Katalogs auf die vorherige Version des Masterimages aus.

- **Die Regeldateien enthalten Syntaxfehler. Weitere Informationen finden Sie in den Protokollen.**

Fehlercode 2. Die Regeldatei enthält Syntaxfehler. Die Personal vDisk-Protokolldatei enthält den Namen der Regeldatei und die Zeile, in der der Syntaxfehler gefunden wurde. Beheben Sie den Syntaxfehler in der Regeldatei und wiederholen Sie den Vorgang.

- **Der in Personal vDisk gespeicherte Bestand, der der vorherigen Version des Masterimages entspricht, ist beschädigt oder nicht lesbar.**

Fehlercode 3. Der letzte Bestand ist in "UserData.V2.vhd" unter "\ProgramData\CitrixPvD\Settings\Inventory\VER-LAST" gespeichert. Stellen Sie den Bestand entsprechend der letzten Version des Masterimages wieder her, indem Sie den Ordner "VER-LAST" von einer funktionierenden PvD-Maschine importieren, die der vorherigen Version des Masterimages zugeordnet ist.

- **Der in Personal vDisk gespeicherte Bestand, der der vorherigen Version des Masterimages entspricht, ist eine höhere Version.**

Fehlercode 4. Die Ursache ist eine PvD-Versionsinkompatibilität zwischen dem letzten Masterimage und dem aktuellen Masterimage. Installieren Sie die aktuelle Version von Personal vDisk auf dem Masterimage und aktualisieren Sie den Katalog erneut.

- **Änderungsjournalüberlauf erkannt.**

Fehlercode 5. Ein USN-Journalüberlauf wurde durch zahlreiche Änderungen am Masterimage während der Bestandserstellung verursacht. Wenn dieses Problem nach mehreren Versuchen weiterhin auftritt, ermitteln Sie mit Process Monitor, ob die Drittanbietersoftware während der Erstellung des Bestands zahlreiche Dateien erstellt oder löscht.

- **Personal vDisk konnte keinen an das System angeschlossenen Datenträger zum Speichern von Benutzerdaten finden.**

Fehlercode 6. Überprüfen Sie zunächst, ob die PvD über die Hypervisor-Konsole mit der VM verbunden ist. Dieser Fehler tritt häufig auf, weil Software zum Vermeiden von Datenverlust den Zugriff auf die PvD verhindert. Wenn die PvD mit der VM verbunden ist, fügen Sie eine Ausnahme für den verbundenen Datenträger in der Konfiguration der Software zum Vermeiden von Datenverlust hinzu.

- **Das System wurde nach der Installation noch nicht neu gestartet. Starten Sie es neu, damit die Änderungen übernommen werden.**

Fehlercode 7. Starten Sie den Desktop neu und wiederholen Sie den Vorgang.

- **Beschädigte Installation. Installieren Sie Personal vDisk neu.**

Fehlercode 8. Installieren Sie Personal vDisk neu und versuchen Sie es noch einmal.

- **Der Personal vDisk-Bestand ist nicht auf dem aktuellen Stand. Aktualisieren Sie den Bestand auf dem Masterimage und versuchen Sie es noch einmal.**

Fehlercode 9. Der Personal vDisk-Bestand wurde vor dem Herunterfahren des Desktops nicht auf dem Masterimages aktualisiert. Starten Sie das Masterimage neu und fahren Sie den Desktop mit der Option "Persönliche vDisk aktualisieren" herunter. Erstellen Sie dann einen neuen Snapshot und aktualisieren Sie damit den Katalog.

- **Beim Start von Personal vDisk ist ein interner Fehler aufgetreten Weitere Informationen finden Sie in den Personal vDisk-Protokollen.**

Fehlercode 10. Die Ursache ist möglicherweise, dass der PvD-Treiber aufgrund eines internen Fehlers oder Beschädigung der persönlichen vDisk keine Virtualisierungssitzung starten kann. Starten Sie den Desktop über den Controller neu. Wenn das Problem weiterhin auftritt, sammeln Sie die Protokolle und wenden Sie sich an den Citrix Support.

- **Personal vDisk-Timeout bei dem Versuch, ein Speichermedium für die Personalisierungseinstellungen der Benutzer zu finden.**

Fehlercode 11. Dieser Fehler tritt auf, wenn der PvD-Treiber den PvD-Datenträger nicht innerhalb von 30 Sekunden nach dem Neustart findet. Die Ursache ist normalerweise ein nicht unterstützter SCSI-Controllertyp oder Speicherlatenz. Wenn dieses Problem bei allen Desktops im Katalog auftritt, ändern Sie den der Vorlagen-VM bzw. Master-VM zugeordneten SCSI-Controllertyp in einen Typ, der von Personal vDisk unterstützt wird. Wenn dieses Problem nur bei einigen Desktops im Katalog auftritt, wird dies möglicherweise durch Speicherlatenz-Spitzen verursacht, weil zahlreiche Desktops gleichzeitig gestartet werden. Beschränken Sie die Einstellung für die maximale Anzahl aktiver Energieaktionen für die Hostverbindung.

- **Personal vDisk wurde deaktiviert, da das System nicht sicher heruntergefahren wurde. Starten Sie die Maschine neu.**

Fehlercode 12. Möglicherweise kann ein Desktop bei aktivierter PvD den Startvorgang nicht abschließen. Starten Sie den Desktop neu. Wenn das Problem weiterhin auftritt, beobachten Sie den Startvorgang des Desktops über die Hypervisor-Konsole und prüfen Sie, ob der Desktop abstürzt. Wenn ein Desktop während des Starts abstürzt, stellen Sie die PvD aus einem Backup (wenn vorhanden) wieder her oder setzen Sie die PvD zurück.

- Der zum Bereitstellen von Personal vDisk angegebene Laufwerksbuchstabe ist nicht verfügbar.**
 Fehlercode 13. Die Ursache ist möglicherweise, dass Personal vDisk den PvD-Datenträger nicht über das vom Administrator angegebene Laufwerk bereitstellen kann. Der PvD-Datenträger kann nicht bereitgestellt werden, wenn der Laufwerksbuchstabe bereits von anderer Hardware verwendet wird. Wählen Sie einen anderen Buchstaben als Bereitstellungspunkt für die persönliche vDisk aus.
- Fehler beim Installieren von Personal vDisk-Kernelmodustreibern.**
 Fehlercode 14. Personal vDisk installiert Treiber während der ersten Bestandsaktualisierung nach der Installation. Einige Antivirenprodukte verhindern die Installation von Treibern außerhalb eines Installationsprogramms. Deaktivieren Sie vorübergehend den in Echtzeit ausgeführten Antivirensan oder fügen Sie der Antivirensoftware während der ersten Bestandserstellung Ausnahmen für die PvD-Treiber hinzu.
- Es konnte kein Snapshot des Systemvolumes erstellt werden. Stellen Sie sicher, dass der Volumeschattenkopie-Dienst aktiviert ist.**
 Fehlercode 15. Dieser Fehler kann auftreten, weil der Volumeschattenkopie-Dienst deaktiviert ist. Aktivieren Sie den Volumeschattenkopie-Dienst und versuchen Sie noch einmal, den Bestand aufzunehmen.
- Fehler beim Aktivieren des Änderungslogs. Warten Sie einige Minuten und versuchen Sie es noch einmal.**
 Fehlercode 16. Personal vDisk verwendet das Änderungslog für das Verfolgen von Änderungen am Masterimage. Wenn PvD bei einer Bestandsaktualisierung erkennt, dass das Änderungslog deaktiviert ist, versucht PvD, es zu aktivieren. Der Fehler tritt auf, wenn die Aktivierung fehlschlägt. Warten Sie ein paar Minuten und versuchen Sie es noch einmal.
- Nicht genügend freier Speicherplatz auf dem Systemvolume.**
 Fehlercode 17. Für das Imageupdate ist nicht genug freier Speicherplatz auf dem Laufwerk C des Desktops vorhanden. Erweitern Sie das Systemvolume oder entfernen Sie nicht verwendete Dateien vom Systemvolume. Das Imageupdate sollte nach dem nächsten Neustart erneut beginnen.
- Es ist nicht genügend freier Speicherplatz im Personal vDisk-Speicher. Erweitern Sie den Personal vDisk-Speicher.**
 Fehlercode 18. Es ist nicht genug freier Speicherplatz auf dem Laufwerk der persönlichen vDisk während des Imageupdatevorgangs vorhanden. Erweitern Sie den Personal vDisk-Speicher oder entfernen Sie nicht verwendete Dateien aus dem Personal vDisk-Speicher. Das Imageupdate sollte nach dem nächsten Neustart erneut beginnen.
- Der Personal vDisk-Speicher ist überbucht. Erweitern Sie den Personal vDisk-Speicher.**
 Fehlercode 19. Es ist nicht genug freier Speicherplatz auf dem Laufwerk der persönlichen vDisk für die in vollem Umfang bereitgestellte "UserData.V2.vhd" vorhanden. Erweitern Sie den Personal vDisk-Speicher oder entfernen Sie nicht verwendete Dateien aus dem Personal vDisk-Speicher.
- Fehlerhafte Systemregistrierung.**
 Fehlercode 20. Die Systemregistrierung ist beschädigt, nicht lesbar oder fehlt. Setzen Sie die persönliche vDisk zurück oder stellen Sie sie von einem früheren Backup wieder her.
- Beim Zurücksetzen von Personal vDisk ist ein interner Fehler aufgetreten. Weitere Informationen finden Sie in den Personal vDisk-Protokollen.**
 Fehlercode 21. Dieser Code gilt für alle Fehler, die beim Zurücksetzen von Personal vDisk auftreten. Sammeln Sie die Protokolle und wenden Sie sich an den Citrix Support.
- Personal vDisk konnte nicht zurückgesetzt werden, da nicht genügend freier Speicherplatz im persönlichen vDisk-Speicher ist.**

Fehlercode 22. Es ist nicht genug freier Speicherplatz auf dem Laufwerk der persönlichen vDisk während eines Zurücksetzungsvorgangs vorhanden. Erweitern Sie den Personal vDisk-Speicher oder entfernen Sie nicht verwendete Dateien aus dem Personal vDisk-Speicher.

Fehlermeldungen: vor Version 7.6

Die folgenden Fehler gelten nur für PvD 7.x-Versionen vor Version 7.6:

- **Start fehlgeschlagen. Personal vDisk konnte kein Speichermedium für die Personalisierungseinstellungen der Benutzer finden.**

Die PvD-Software konnte die Personal vDisk (standardmäßig Laufwerk P:) nicht finden oder nicht als den vom Administrator bei der Erstellung des Katalogs ausgewählten Bereitstellungspunkt bereitstellen.

- Prüfen Sie, ob Sie im PvD-Dienstprotokoll den folgenden Eintrag finden: "PvD 1 status --> 18:183".
 - Wenn Sie eine ältere Version als PvD Version 5.6.12 verwenden, löst ein Upgrade auf die aktuelle Version das Problem.
 - Wenn Sie Version 5.6.12 oder höher verwenden, prüfen Sie mit dem Datenträgerverwaltungstool (diskmgmt.msc), ob das Laufwerk P: als nicht bereitgestelltes Volume vorhanden ist. Wenn es vorhanden ist, führen Sie chkdsk auf dem Volume aus, um festzustellen, ob es fehlerhaft ist. Versuchen Sie, das Volume mit chkdsk wiederherzustellen.
- **Start fehlgeschlagen. Fehler beim Start von Citrix Personal vDisk. Weitere Informationen ... Statuscode: 7, Fehlercode: 0x70**

Statuscode 7 bedeutet, dass ein Fehler beim Update der PvD aufgetreten ist. Es gibt die folgenden Fehler:

Fehlercode	Beschreibung
0x20000001	Fehler beim Speichern des Vergleichspakets, wahrscheinlich aufgrund von Speicherplatzmangel auf der virtuellen Festplatte.
0x20000004	Unzureichende Privilegien zum Aktualisieren der PvD.
0x20000006	Fehler beim Laden der Struktur aus dem PvD-Image oder dem PvD-Bestand, wahrscheinlich aufgrund eines fehlerhaften PvD-Images oder PvD-Bestands.
0x20000007	Fehler beim Laden des Dateisystembestands, wahrscheinlich aufgrund eines fehlerhaften PvD-Images oder PvD-Bestands.
0x20000009	Fehler beim Öffnen der Datei mit dem Dateisystembestand, wahrscheinlich aufgrund eines fehlerhaften PvD-Images oder PvD-Bestands.
0x2000000B	Fehler beim Speichern des Vergleichspakets, wahrscheinlich aufgrund von Speicherplatzmangel auf der virtuellen Festplatte.
0x20000010	Fehler beim Laden des Vergleichspakets.
0x20000011	Regeldateien fehlen.
0x20000021	Fehlerhafter PvD-Bestand.
0x20000027	Der Katalog "MojoControl.dat" ist fehlerhaft.

0x2000002B Fehlercode	PvD-Bestand fehlerhaft oder fehlt. Beschreibung
0x2000002F	Fehler beim Registrieren des vom Benutzer installierten MOF beim Imageupdate. Lösung: Upgrade auf Version 5.6.12.
0x20000032	Suchen Sie in PvDactivation.log.txt nach dem letzten Eintrag mit einem Win32-Fehlercode.
0x20	Fehler beim Bereitstellen des Anwendungscontainers für Imageupdate. Lösung: Upgrade auf Version 5.6.12.
0x70	Nicht genügend Speicherplatz auf dem Datenträger.

- **Start fehlgeschlagen. Fehler beim Start von Citrix Personal vDisk [oder Personal vDisk hat einen internen Fehler festgestellt]. Weitere Informationen... Statuscode 20, Fehlercode 0x20000028**

Die persönliche vDisk wurde gefunden, eine PvD-Sitzung konnte jedoch nicht erstellt werden.

Sammeln Sie die Protokolle und prüfen Sie das Protokoll "SysVol-IvmSupervisor.log" auf Sitzungserstellungsfehler:

1. Suchen Sie nach dem Protokolleintrag "IvmpNativeSessionCreate: failed to create native session, status XXXXX".
2. Wenn der Status 0xc00002cf ist, können Sie das Problem lösen, indem Sie eine neue Version des Masterimages zum Katalog hinzufügen. Dieser Statuscode bedeutet, dass ein USN-Journalüberlauf aufgrund zahlreicher Änderungen nach einer Bestandsaktualisierung aufgetreten ist.
3. Starten Sie den betroffenen virtuellen Desktop neu. Wenn das Problem weiterhin besteht, wenden Sie sich an den technischen Support von Citrix.

- **Start fehlgeschlagen. Citrix Personal vDisk wurde deaktiviert, da das System nicht sicher heruntergefahren wurde. Wählen Sie "Noch einmal versuchen". Wenn das Problem weiterhin besteht, wenden Sie sich an den Systemadministrator.**

Die gepoolte VM kann den Start nicht ausführen, solange die PvD aktiviert ist. Stellen Sie zuerst fest, warum der Start nicht ausgeführt werden kann. Eine mögliche Ursache ist die Anzeige eines blauen Bildschirms aus einem der folgenden Gründe:

- Ein nicht kompatibles Antivirenprodukt ist auf dem Masterimage vorhanden, z. B. alte Versionen von Trend Micro.
- Der Benutzer hat Software installiert, die mit PvD nicht kompatibel ist. Dies ist zwar unwahrscheinlich, aber Sie können es überprüfen, indem Sie dem Katalog eine neue Maschine hinzufügen und testen, ob sie neu startet.
- Das PvD-Image ist fehlerhaft. Dieser Fehler wurde in Version 5.6.5 beobachtet.

So prüfen Sie, ob die gepoolte VM einen blauen Bildschirm anzeigt oder ob sie vorzeitig neu startet:

- Melden Sie sich bei der Maschine über die Hypervisor-Konsole an.
- Klicken Sie auf Noch einmal versuchen und warten Sie, bis die Maschine heruntergefahren ist.
- Starten Sie die Maschine über Studio.
- Beobachten Sie die Maschinenkonsole während des Starts mit der Hypervisor-Konsole.

Weitere Problembeschreibungsschritte:

- Senden Sie das Speicherabbild der Maschine, die den blauen Bildschirm hat, zur Analyse an den technischen Support von Citrix.
- Prüfen Sie auch die der PvD zugeordneten Ereignisprotokolle auf Fehler:
 1. Stellen Sie "UserData.V2.vhd" (im Stamm von Laufwerk P:) mithilfe von DiskMgmt.msc bereit, indem Sie auf Aktion > Virtuelle Festplatte anfügen klicken.
 2. Starten Sie "Eventvwr.msc".
 3. Öffnen Sie das Systemereignisprotokoll (Windows\System32\winevt\logs\system.evtx) aus UserData.V2.vhd, indem

Sie auf Aktion > Gespeichertes Protokoll öffnen klicken.

- Öffnen Sie das Anwendungsereignisprotokoll (Windows\System32\winevt\logs\application.evtx) aus UserData.V2.vhd, indem Sie auf Aktion > Gespeichertes Protokoll öffnen klicken.

- Die persönliche vDisk kann nicht gestartet werden. Die persönliche vDisk konnte nicht gestartet werden, da der Bestand nicht aktualisiert worden ist. Aktualisieren Sie den Bestand auf dem Masterimage und versuchen Sie es noch einmal. Statuscode: 15, Fehlercode: 0x0**

Der Administrator hat beim Erstellen oder Aktualisieren des PvD-Katalogs den falschen Snapshot ausgewählt (d. h. das Masterimage wurde beim Erstellen des Snapshots nicht mit Persönliche vDisk aktualisieren heruntergefahren).

Von Personal vDisk protokollierte Ereignisse

Wenn Personal vDisk nicht aktiviert ist, können Sie die folgenden Ereignisse in der Windows-Ereignisanzeige anzeigen. Wählen Sie im linken Bereich den Knoten Anwendungen, die Quelle im rechten Bereich ist Citrix Personal vDisk. Wenn Personal vDisk aktiviert ist, werden diese Ereignisse nicht angezeigt.

Die Ereignis-ID 1 bedeutet, dass es sich um eine Informationsmeldung handelt. ID 2 steht für einen Fehler. Möglicherweise werden nicht alle Ereignisse in jeder Personal vDisk-Version verwendet.

Ereignis-ID	Beschreibung
1	Status von Personal vDisk: Bestandsaktualisierung gestartet.
1	Status von Personal vDisk: Bestandsaktualisierung abgeschlossen. GUID: %s.
1	Status von Personal vDisk: Imageupdate gestartet.
1	Status von Personal vDisk: Imageupdate abgeschlossen.
1	Zurücksetzen wird durchgeführt.
1	OK.
2	Status von Personal vDisk: Bestandsaktualisierung fehlgeschlagen mit: %s.
2	Status von Personal vDisk: Imageupdate fehlgeschlagen mit: %s.
2	Status von Personal vDisk: Imageupdate fehlgeschlagen, interner Fehler.
2	Status von Personal vDisk: Bestandsaktualisierung fehlgeschlagen: interner Fehler.
2	Personal vDisk wurde unsachgemäß beendet und daher deaktiviert.
2	Imageupdate fehlgeschlagen. Fehlercode %d.

Ereignis-ID	Beschreibung Personal vDisk hat einen internen Fehler festgestellt. Statuscode[%d] Fehlercode[0x%X].
2	Zurücksetzung von Personal vDisk fehlgeschlagen.
2	Datenträger zum Speichern der angepassten Benutzereinstellungen kann nicht gefunden werden.
2	Zum Erstellen eines Personal vDisk-Containers ist nicht genug Speicherplatz auf dem Datenträger vorhanden.

Konfigurationsprotokollierung

Sep 29, 2015

Die Konfigurationsprotokollierung dient zum Erfassen der Sitekonfigurationsänderungen und Administratoraktivitäten in einer Datenbank. Sie können den protokollierten Inhalt folgendermaßen verwenden:

- Diagnose und Problembehandlung nach der Durchführung von Konfigurationsänderungen; das Protokoll liefert eine Breadcrumbspur.
- Hilfe beim Änderungsmanagement und der Nachverfolgung von Konfigurationen
- Bericht über Administratoraktivitäten

Zum Festlegen der Einstellungen für die Konfigurationsprotokollierung, zum Anzeigen der Konfigurationsprotokolle und zum Generieren von HTML- und CSV-Berichten verwenden Sie Citrix Studio. Sie können die Anzeige des Konfigurationsprotokolls durch Datumsbereiche und durch Ergebnisse der Volltextsuche filtern. Ist die verbindliche Protokollierung aktiviert, verhindert sie, dass Änderungen an der Konfiguration vorgenommen werden, es sei denn diese können protokolliert werden. Mit der entsprechenden Berechtigung können Sie Einträge aus dem Konfigurationsprotokoll löschen. Sie können das Feature der Konfigurationsprotokollierung nicht zum Bearbeiten des Inhalts von Protokollen verwenden.

Die Konfigurationsprotokollierung verwendet ein PowerShell 2.0-SDK und den Konfigurationsprotokollierungsdienst. Der Konfigurationsprotokollierungsdienst wird auf allen Controllern in der Site ausgeführt. Wenn ein Controller ausfällt, übernimmt automatisch der Dienst auf einem anderen Controller die Verarbeitung von Protokollanforderungen.

Standardmäßig ist die Konfigurationsprotokollierung aktiviert und verwendet die Datenbank, die zusammen mit der Site erstellt wurde (die Sitekonfigurationsdatenbank). Citrix empfiehlt dringend, den Speicherort der Datenbank für die Konfigurationsprotokollierung so schnell wie möglich nach der Erstellung einer Site zu ändern. Die Konfigurationsprotokollierungsdatenbank unterstützt dieselben Features für hohe Verfügbarkeit wie die Sitekonfigurationsdatenbank.

Der Zugriff auf die Konfigurationsprotokollierung wird über die delegierte Administration mit den Einstellungen "Protokollierungseinstellungen bearbeiten" und "Konfigurationsprotokolle anzeigen" gesteuert.

Konfigurationsprotokolle werden bei der Erstellung lokalisiert. Beispiel: Ein in Englisch erstelltes Protokoll wird unabhängig vom Gebietsschema des Lesers in Englisch gelesen.

Gegenstand der Protokollierung

Konfigurationsänderungen und Administratoraktivitäten, die von Studio, Director und PowerShell-Skripts ausgehen, werden protokolliert. Beispiele protokollierter Konfigurationsänderungen sind Arbeiten (Erstellen, Bearbeiten, Löschen, Zuweisen) mit:

- Maschinenkataloge
- Bereitstellungsgruppen (einschließlich Ändern der Energieverwaltungseinstellungen)
- Administratorrollen und Geltungsbereiche
- Hostressourcen und Verbindungen
- Citrix Richtlinien über Studio

Beispiele protokollierter Administratoraktivitäten:

- Energieverwaltung für eine virtuelle Maschine oder einen Benutzerdesktop
- Senden einer Nachricht an einen Benutzer von Studio oder Director aus

Die folgenden Vorgänge werden nicht protokolliert:

- Autonome Vorgänge wie das Einschalten virtueller Maschinen per Poolverwaltung.
- Über die Gruppenrichtlinien-Verwaltungskonsolle implementierte Richtlinienaktionen; verwenden Sie Microsoft-Tools, um Protokolle dieser Aktionen anzuzeigen.
- Über die Registrierung vorgenommene Änderungen, direkter Zugriff von der Datenbank oder von anderen Quellen als Studio, Director oder PowerShell.
- Wenn die Bereitstellung initialisiert wird, steht die Konfigurationsprotokollierung ab dem Zeitpunkt zur Verfügung, zu dem die erste Instanz des Konfigurationsprotokollierungsdiensts sich beim Konfigurationsdienst registriert. Daher werden die frühen Phasen der Konfiguration nicht protokolliert (z. B., wenn das Datenbankschema bei der Initialisierung eines Hypervisors abgerufen und angewendet wird).

Verwalten der Konfigurationsprotokollierung

Standardmäßig ist die Konfigurationsprotokollierung aktiviert und die verbindliche Protokollierung ist deaktiviert.

Standardmäßig wird für die Konfigurationsprotokollierung die Datenbank verwendet, die zusammen mit einer Site erstellt wird (die Sitekonfigurationsdatenbank). Citrix empfiehlt, den Speicherort der Datenbank für die Konfigurationsprotokollierung (und der Datenbank für den Überwachungsdienst, von dem die Sitekonfigurationsdatenbank auch standardmäßig verwendet wird) aus folgenden Gründen nach der Erstellung einer Site zu ändern:

- Die Backupstrategie für die Konfigurationsprotokollierungsdatenbank unterscheidet sich wahrscheinlich von der Backupstrategie für die Sitekonfigurationsdatenbank.
- Die Menge der für die Konfigurationsprotokollierung (und den Überwachungsdienst) gesammelten Daten kann den für die Sitekonfigurationsdatenbank verfügbaren Speicherplatz zu stark limitieren.
- Eine einzelne Fehlerquelle für die drei Datenbanken wird beseitigt (d. h. aufgeteilt).

Weitere Informationen finden Sie unter [Ändern des Speicherorts der sekundären Datenbanken](#).

Die Konfigurationsprotokollierungsdatenbank unterstützt dieselben Features für hohe Verfügbarkeit wie die Sitekonfigurationsdatenbank. Lesen Sie den Leitfaden zu Datenbanken unter [Planen der Bereitstellung](#).

Aktivieren/Deaktivieren der Konfigurationsprotokollierung und der verbindlichen Protokollierung

1. Wählen Sie in Citrix Studio im linken Bereich Protokollierung aus.
Hinweis: Editionen, die keine Konfigurationsprotokollierung unterstützen, haben keinen Knoten namens Protokollierung.
2. Klicken Sie im Bereich "Aktionen" auf Einstellungen. Das Dialogfeld Konfigurationsprotokollierung wird geöffnet. Dieses enthält die Datenbankinformationen und Angaben dazu, ob Konfigurationsprotokollierung und verbindliche Protokollierung aktiviert oder deaktiviert sind.
 - Aktivieren oder Deaktivieren der Konfigurationsprotokollierung
 - Zum Aktivieren der Konfigurationsprotokollierung wählen Sie das Optionsfeld Protokollierung aktivieren. Dies ist die Standardeinstellung. Wenn nicht in die Datenbank geschrieben werden kann, werden die Informationen verworfen, der Vorgang wird jedoch fortgesetzt.
 - Zum Deaktivieren der Konfigurationsprotokollierung wählen Sie das Optionsfeld Protokollierung deaktivieren. Wenn die Protokollierung zuvor aktiviert war, können bereits vorhandene Protokolle weiterhin mit dem PowerShell-SDK gelesen werden.
 - Aktivieren oder Deaktivieren der verbindlichen Protokollierung
 - Zum Aktivieren der verbindlichen Protokollierung deaktivieren Sie das Kontrollkästchen Änderungen zulassen, wenn die Verbindung mit der Datenbank getrennt ist. Es wird dann keine Konfigurationsänderung oder administrative Aktivität, die normalerweise protokolliert würde, zugelassen, es sei denn, sie kann in die Datenbank für die

Konfigurationsprotokollierung geschrieben werden.

- Zum Deaktivieren der verbindlichen Protokollierung aktivieren Sie das Kontrollkästchen Änderungen zulassen, wenn die Verbindung mit der Datenbank getrennt ist. Konfigurationsänderungen und administrative Aktivitäten sind dann zulässig, selbst wenn kein Zugriff auf die Datenbank für die Konfigurationsprotokollierung besteht. Dies ist die Standardeinstellung.

Die Option zur Steuerung der verbindlichen Protokollierung ist nur verfügbar, wenn die Konfigurationsprotokollierung aktiviert ist, d. h. wenn das Optionsfeld Protokollierung aktivieren für die Konfigurationsprotokollierung aktiviert ist. Tritt bei dem Dienst für die Konfigurationsprotokollierung ein Fehler auf, und die hohe Verfügbarkeit wird nicht verwendet, beginnt die verbindliche Protokollierung. In solchen Fällen werden Vorgänge, die normalerweise protokolliert würden, nicht ausgeführt.

Ändern des Speicherorts für die Konfigurationsprotokollierungsdatenbank

1. Erstellen Sie einen Datenbankserver mit einer unterstützten SQL Server-Version.
2. Wählen Sie in Studio im linken Bereich Protokollierung aus.
3. Klicken Sie im Bereich "Aktionen" auf Einstellungen. Das Dialogfeld Konfigurationsprotokollierung wird angezeigt.
4. Klicken Sie auf Protokollierungsdatenbank ändern. Das Dialogfeld Protokollierungsdatenbank ändern wird angezeigt.
5. Geben Sie mit einem der Formulare aus der folgenden Tabelle den Speicherort des Servers ein, auf dem sich der neue Datenbankserver befindet, und den Namen der Datenbank.

Datenbanktyp	Eingabe	Auswirkungen dieser Datenbankkonfiguration
Eigenständig oder gespiegelt	servername	Die Standardinstanz wird verwendet und SQL Server verwendet den Standardport.
	Servername\Instanzname	Eine benannte Instanz wird verwendet und SQL Server verwendet den Standardport.
	servername,Portnummer	Die Standardinstanz wird verwendet und SQL Server verwendet einen benutzerdefinierten Port. (Das Komma ist erforderlich.)
Sonstiges	cluster-name	Eine geclusterte Datenbank.
	availability-group-listener	Eine stets aktivierte Datenbank.

6. Wenn die Datenbank von Studio erstellt werden soll, klicken Sie auf OK oder Verbindung testen. Wenn Sie dazu aufgefordert werden, klicken Sie auf OK und Studio erstellt die Datenbank automatisch. Studio versucht, mit den Anmeldeinformationen des aktuellen Studio-Benutzers auf die Datenbank zuzugreifen; wenn dies fehlschlägt, werden Sie zur Eingabe der Anmeldeinformationen des Datenbankbenutzers aufgefordert. Das Datenbankschema wird dann von Studio in die Datenbank hochgeladen. (Die Anmeldeinformationen werden nur für den Zeitraum der Datenbankerstellung gespeichert.)
7. Wenn Sie die Datenbank manuell erstellen möchten, klicken Sie auf Skript erstellen (oder verwenden Sie Get-LogDBSchema im PowerShell SDK). Das generierte Skript enthält Anweisungen zum manuellen Erstellen der Datenbank. Stellen Sie vor dem Hochladen des Schemas sicher, dass die Datenbank leer ist und dass mindestens ein Benutzer Zugriffs- bzw. Änderungsberechtigung für die Datenbank hat.

Die Daten der Konfigurationsprotokollierung aus der älteren Datenbank werden nicht in die neue Datenbank importiert. Die

Protokolle beider Datenbanken können beim Abrufen von Protokollen nicht aggregiert werden. Der erste Protokolleintrag in der neuen Datenbank für die Konfigurationsprotokollierung gibt an, dass eine Datenbankänderung stattfand, die vorherige Datenbank wird jedoch nicht identifiziert.

Anzeigen des Konfigurationsprotokolls

Beim Initiieren von Konfigurationsänderungen und Administratoraktivitäten werden in Citrix Studio und Citrix Director — *High-Level-Operationen* erstellt, die im oberen Teil des mittleren Bereichs in Studio angezeigt werden. Eine High-Level-Operation führt zu mindestens einem Dienst- und SDK-Aufruf, bei dem es sich um eine — *Low-Level-Operation* handelt. Wenn Sie eine High-Level-Operation im oberen Teil des mittleren Bereichs auswählen, werden im unteren Teil des mittleren Bereichs die Low-Level-Operationen angezeigt.

Schlägt eine Operation vor der Beendigung fehl, kann die Protokollierung evtl. in der Datenbank nicht abgeschlossen werden, sodass z. B. ein Startdatensatz keinen entsprechenden Stoppsatz hat. In solchen Fällen wird im Protokoll angezeigt, dass Informationen fehlen. Wenn Sie Protokolle auf Zeitbereichsbasis anzeigen, werden unvollständige Protokolle angezeigt, wenn die Daten in den Protokollen mit den Kriterien übereinstimmen. Beispiel: Wenn alle Protokolle für die letzten fünf Tage angefordert werden und ein Protokoll eine in den letzten fünf Tagen gelegene Startzeit aber keine Endzeit hat, wird dieses ebenfalls angezeigt.

Wenn Sie bei Verwendung eines Skripts zum Aufruf von PowerShell-Cmdlets eine Low-Level-Operation erstellen ohne die übergeordnete High-Level-Operation anzugeben, wird von der Konfigurationsprotokollierung eine Ersatz-High-Level-Operation erstellt.

Anzeigen des Konfigurationsprotokolls

Wählen Sie in Studio im linken Bereich Protokollierung aus. Standardmäßig wird im mittleren Bereich der Protokollinhalt chronologisch (neueste Einträge zuerst), angezeigt, wobei die Einträge durch das Datum getrennt sind.

Zum Filtern der Anzeige nach folgenden Kriterium	Führen Sie folgenden Schritt aus
Suchergebnisse	Geben Sie Text in das Feld Suchen oben im mittleren Bereich ein. Die gefilterte Anzeige enthält die Anzahl der Suchergebnisse. Zum Zurückkehren zur Standardprotokollanzeige löschen Sie den Text im Feld Suchen.
Spaltenüberschrift	Klicken Sie auf eine Spaltenüberschrift, um die Anzeige nach dem entsprechenden Feld zu sortieren.
Datumsbereich	Wählen Sie einen Zeitraum aus der Dropdownliste neben dem Feld Suchen oben im mittleren Bereich aus (Heute, Letzte 7 Tage, Letzte 28 Tage, Letzte drei Monate, Letzte sechs Monate).

Generieren von Konfigurationsprotokollberichten

Sie können CSV- und HTML-Berichte mit Konfigurationsprotokolldaten generieren.

- Der CSV-Bericht ist ein Abbild der Protokolldaten aus einem angegebenen Zeitintervall. Die hierarchischen Daten in der Datenbank werden in eine einzelne CSV-Tabelle vereinfacht. Kein Aspekt der Daten hat Vorrang in der Datei. Es wird keine Formatierung verwendet und keine Lesbarkeit angenommen. Die Datei (unter dem Namen "MyReport") enthält

lediglich die Daten in einem allgemein verwendbaren Format. CSV-Dateien werden oft für die Archivierung oder als Datenquelle für ein Tool zur Bearbeitung von Berichten oder Daten (z. B. Microsoft Excel) verwendet.

- Der HTML-Bericht enthält Protokolldaten aus einem angegebenen Zeitintervall in lesbarem Format. Er bietet eine strukturierte Ansicht für die Prüfung auf Änderungen, durch die navigiert werden kann. Der HTML-Bericht umfasst zwei Dateien: Zusammenfassung und Details. Die Zusammenfassung enthält High-Level-Operationen mit Informationen zu Zeitpunkt, Auslöser und Ergebnis. Durch Klicken auf den Link Details neben jedem Vorgang navigieren Sie zu den Low-Level-Operationen in der Datei Details, die zusätzliche Informationen bietet.

Generieren eines Konfigurationsprotokollberichts

1. Wählen Sie in Citrix Studio im linken Bereich Protokollierung aus.
2. Klicken Sie im Bereich "Aktionen" auf Benutzerdefinierten Bericht erstellen.
3. Wählen Sie den Datumsbereich für den Bericht, indem Sie ein vordefiniertes Intervall auswählen oder einen benutzerdefinierten Zeitrahmen eingeben.
4. Wählen Sie das Berichtsformat: CSV, HTML oder Beide.
5. Navigieren Sie zu dem Speicherort, an dem der Bericht gespeichert werden soll.
6. Prüfen Sie Ihre Auswahl auf der Seite Zusammenfassung und klicken Sie auf Fertig stellen.

Löschen des Konfigurationsprotokolls

Zum Löschen des Konfigurationsprotokolls müssen Sie über bestimmte Rechte der delegierten Administration und Berechtigungen für die SQL Server-Datenbank verfügen.

- **Delegierte Administration:** Sie müssen eine Rolle der delegierten Administration haben, mit der die Bereitstellungsconfiguration gelesen werden kann. Die integrierte Volladministratorrolle hat diese Berechtigung. Für eine benutzerdefinierte Rolle muss für die Kategorie "Andere Berechtigungen" "Lesen" oder "Verwalten" aktiviert sein. Wenn Sie die Konfigurationsprotokolldaten vor dem Löschen sichern möchten, muss die benutzerdefinierte Rolle in der Kategorie der Protokollierungsberechtigungen Lese- oder Verwaltungsberechtigung haben.
- **SQL Server-Datenbank:** Sie müssen einen Anmeldenamen für SQL Server haben und zum Löschen von Datensätzen aus der Datenbank berechtigt sein. Dies kann mit zwei Möglichkeiten erreicht werden:
 - Verwenden Sie zur Anmeldung für die SQL Server-Datenbank die Serverrolle "sysadmin", mit der Sie beliebige Aktivitäten auf dem Datenbankserver durchführen können. Auch die Serverrollen "serveradmin" oder "setupadmin" sind zum Löschen von Vorgängen berechtigt.
 - Wenn Ihre Bereitstellung zusätzliche Sicherheit erfordert, verwenden Sie Anmeldeinformationen einer anderen Rolle als "sysadmin", die einem Datenbankbenutzer zugeordnet sind, der zum Löschen von Datensätzen aus der Datenbank berechtigt ist.
 1. Erstellen Sie in SQL Server Management Studio eine SQL Server-Anmeldung mit einer anderen Serverrolle (nicht "sysadmin").
 2. Ordnen Sie diese Anmeldung einem Benutzer in der Datenbank zu; SQL Server erstellt automatisch einen Benutzer in der Datenbank mit dem gleichen Namen wie die Anmeldung.
 3. Geben Sie für die Datenbankrollen-Mitgliedschaft mindestens eines der Rollenmitglieder für den Datenbankbenutzer an: ConfigurationLoggingSchema_ROLE oder dbowner.Weitere Informationen finden Sie in der Dokumentation zu SQL Server Management Studio.

Löschen der Konfigurationsprotokolle

1. Wählen Sie in Studio im linken Bereich Protokollierung aus.
2. Klicken Sie im Aktionsbereich auf Protokolle löschen.

3. Sie haben nun die Möglichkeit, vor dem Löschen ein Backup der Protokolle anzulegen. Wenn Sie eine Backupdatei erstellen, navigieren Sie zu dem Speicherort, an dem diese gespeichert werden soll. Das Backup wird als CSV-Datei erstellt.
4. Überprüfen Sie Ihre Auswahl auf der Seite Zusammenfassung und klicken Sie dann auf Fertig stellen.

Nach dem Löschen der Konfigurationsprotokolle wird das Löschen des Protokolls als erste Aktivität im leeren Protokoll erfasst. Dieser Eintrag enthält Details darüber, wann und von wem die Protokolle gelöscht wurden.

Delegierte Administration

Sep 29, 2015

Das Modell der delegierten Administration bietet Flexibilität bei der Delegierung der Administratoraktivitäten mit Rollen und der objektbasierten Steuerung. Die delegierte Administration ist für Bereitstellungen aller Größen geeignet und ermöglicht es Ihnen, mit zunehmender Komplexität der Bereitstellung die Berechtigungsgranularität zu erhöhen. Bei der delegierten Administration werden drei Konzepte eingesetzt: Administratoren, Rollen und Geltungsbereiche.

- **Administratoren:** Administratoren sind Einzelpersonen oder Personengruppen, die durch ein Active Directory-Konto identifiziert werden. Jeder Administrator ist mit mindestens einem Paar aus Rolle und Geltungsbereich verknüpft.
- **Rollen:** Eine Rolle repräsentiert eine Stellenfunktion und ist mit definierten Berechtigungen ausgestattet. Beispiel: Die Rolle "Bereitstellungsgruppenadministrator" verfügt über Berechtigungen wie etwa "Bereitstellungsgruppe erstellen" und "Desktop aus Bereitstellungsgruppe entfernen". Ein Administrator kann mehrere Rollen für eine Site haben, d. h. eine Person kann sowohl Bereitstellungsgruppenadministrator als auch Maschinenkatalogadministrator sein. Rollen können integriert oder benutzerdefiniert sein.

Integrierte Rollen:

Rolle	Berechtigungen
Volladministrator	Kann alle Aufgaben und Vorgänge ausführen. Ein Volladministrator wird immer mit dem Geltungsbereich "Alle" kombiniert.
Lesezugriffadministrator	Kann alle Objekte in den angegebenen Geltungsbereichen sowie globale Informationen anzeigen, aber nicht ändern. Beispiel: Ein Lesezugriffadministrator mit Geltungsbereich = London kann alle globalen Objekte (z. B. Konfigurationsprotokollierung) und alle London-bezogenen Geltungsbereichsobjekte (z. B. London-Bereitstellungsgruppen) sehen. Dieser Administrator kann jedoch nicht die Objekte im Geltungsbereich "New York" sehen (sofern die Geltungsbereiche "London" und "New York" einander nicht überlappen).
Helpdeskadministrator	Kann Bereitstellungsgruppen anzeigen und die diesen Gruppen zugeordneten Sitzungen und Maschinen verwalten. Kann den Maschinenkatalog und die Hostinformationen der überwachten Bereitstellungsgruppen sehen sowie Sitzungsverwaltungs- und Energieverwaltungsvorgänge für die Maschinen in diesen Bereitstellungsgruppen durchführen.
Maschinenkatalogadministrator	Kann Maschinenkataloge erstellen und verwalten sowie darin Maschinen bereitstellen. Kann Maschinenkataloge aus der Virtualisierungsinfrastruktur, Provisioning Services und von physischen Maschinen anlegen. Mit dieser Rolle können Basisimages verwaltet und Software installiert werden, aber den Benutzern können keine Anwendungen oder Desktops zugewiesen werden.
Bereitstellungsgruppenadministrator	Kann Anwendungen, Desktops und Maschinen bereitstellen sowie die mit ihnen verbundenen Sitzungen verwalten. Kann zudem Anwendungs- und Desktopkonfigurationen wie Richtlinien und die Energieverwaltungseinstellungen verwalten.

Hostadministrator Rolle	Kann Hostverbindungen und ihnen zugeordnete Ressourceneinstellungen verwalten. Kann keine Maschinen, Anwendungen oder Desktops für Benutzer bereitstellen.
----------------------------	--

Bei bestimmten Produkteditionen können Sie benutzerdefinierte Rollen erstellen, um sie den Anforderungen Ihrer Organisation anzupassen und die Berechtigungen entsprechend delegieren. Sie können benutzerdefinierte Rollen dazu verwenden, Berechtigungen in der Granularität einer Aktion oder Aufgabe in einer Konsole zuzuteilen.

- **Geltungsbereiche:** Ein Geltungsbereich steht für eine Sammlung von Objekten. Geltungsbereiche werden verwendet, um die Objekte in einer für Ihre Organisation angemessenen Weise zu gruppieren (z. B. die Bereitstellungsgruppen der Vertriebsabteilung). Objekte können in mehreren Geltungsbereichen vertreten sein, d. h. Objekte können durch einen oder mehrere Geltungsbereiche bezeichnet sein. Der einzige integrierte Geltungsbereich "Alle" enthält alle Objekte. Die Volladministratorrolle bildet immer ein Paar mit dem Geltungsbereich "Alle".

Firma XYZ entscheidet sich zum Verwalten von Anwendungen und Desktops basierend auf ihrer Abteilungsstruktur (Buchhaltung, Vertrieb und Lager) und ihren Desktopbetriebssystemen (Windows 7 oder Windows 8). Der Administrator erstellt fünf Geltungsbereiche und erfasst jede Bereitstellungsgruppe in zwei Geltungsbereichen: einem Geltungsbereich für die Abteilung, in der sie verwendet werden und einem Geltungsbereich für das verwendete Betriebssystem.

Die folgenden Administratoren wurden erstellt:

Administrator	Rollen	Geltungsbereiche
domain/fred	Volladministrator	Alle (Volladministratorrolle wird immer mit "Alle" ausgestattet)
domain/rob	Lesezugriffadministrator	Alle
domain/heidi	Lesezugriffadministrator Helpdeskadministrator	Alle Vertrieb
domain/warehouseadmin	Helpdeskadministrator	Lager
domain/peter	Bereitstellungsgruppenadministrator und Maschinenkatalogadministrator	Win7

- Fred ist Volladministrator und kann alle Elemente im System anzeigen, bearbeiten und löschen.
- Rob kann alle Objekte der Site anzeigen jedoch nicht bearbeiten oder löschen.
- Heidi kann alle Objekte anzeigen und Helpdeskaufgaben an Bereitstellungsgruppen des Geltungsbereichs "Vertrieb" durchführen. Somit kann sie die diesen Gruppen zugeordneten Sitzungen und Maschinen verwalten; sie kann allerdings keine Änderungen an der Bereitstellungsgruppe durchführen, wie Hinzufügen oder Entfernen von Maschinen.
- Jedes Mitglied der Active Directory-Sicherheitsgruppe "warehouseadmin" kann Helpdeskaufgaben für Maschinen des Geltungsbereichs "Lager" ausführen.

- Peter ist Spezialist für Windows 7 und kann alle Windows 7-Maschinenkataloge verwalten und Windows 7-Anwendungen, -Desktops und -Maschinen bereitstellen, unabhängig davon, in welchem Abteilungsgeltungsbereich sie sich befinden. Der Administrator erwägt, Peter zum Volladministrator für den Geltungsbereich "Win7" zu machen; entscheidet sich jedoch dagegen, da ein Volladministrator ebenfalls über vollständige Administratorrechte für alle Objekte verfügt, die nicht in einen Geltungsbereich fallen, z. B. "Site" und "Administrator".

Im Allgemeinen hängt die Anzahl der Administratoren und die Granularität der Berechtigungen von der Größe und Komplexität der Bereitstellung ab.

- In kleinen Bereitstellungen oder Testbereitstellungen übernehmen ein oder wenige Administratoren alle Aufgaben und es findet keine Delegation statt. Erstellen Sie in diesem Fall einen einzelnen Administrator mit der integrierten Rolle "Volladministrator", die den Geltungsbereich "Alle" hat.
- In größeren Bereitstellungen mit mehr Maschinen, Anwendungen und Desktops ist mehr Delegation erforderlich. Mehrere Administratoren haben möglicherweise bestimmte funktionale Zuständigkeiten (Rollen). Beispiel: Es gibt zwei Volladministratoren, andere sind Helpdeskadministratoren. Weiterhin werden von einem Administrator ggf. nur bestimmte Objektgruppen (Geltungsbereiche) wie Maschinenkataloge verwaltet. Erstellen Sie in diesem Fall neue Geltungsbereiche und Administratoren mit einer der integrierten Rollen und den entsprechenden Geltungsbereichen.
- Noch größere Bereitstellungen erfordern möglicherweise weitere (oder differenziertere) Geltungsbereiche sowie andere Administratoren mit ungewöhnlichen Rollen. Bearbeiten oder erstellen Sie in diesem Fall weitere Geltungsbereiche, erstellen Sie benutzerdefinierte Rollen und erstellen Sie jeden Administrator mit einer integrierten oder benutzerdefinierten Rolle sowie vorhandenen und neuen Geltungsbereichen.

Für mehr Flexibilität und zur Vereinfachung der Konfiguration können Sie neue Geltungsbereiche erstellen, wenn Sie einen Administrator erstellen. Sie können auch beim Erstellen oder Bearbeiten von Maschinenkatalogen oder Hosts Geltungsbereiche festlegen.

Administratoren

Sep 29, 2015

Beim Erstellen einer Site als lokaler Administrator wird dieses Benutzerkonto automatisch zum Volladministrator mit Vollzugriff auf alle Objekte. Nachdem die Site erstellt wurde, verfügen lokale Administratoren über keine besonderen Rechte.

Standardmäßig wird ein Administrator aktiviert. Sie können den Administrator bei Bedarf deaktivieren. Dies kann erforderlich sein, wenn Sie einen neuen Administrator konfigurieren, diese Person die Administratortaufgaben jedoch erst zu einem späteren Zeitpunkt wahrnehmen wird. Bei vorhandenen aktivierten Administratoren kann es vorkommen, dass Sie einige deaktivieren müssen, während Sie Objekte/Geltungsbereiche neu strukturieren und sie dann wieder aktivieren, wenn Sie die Aktualisierung der Konfiguration abgeschlossen haben.

Wichtig: Um zu verhindern, dass das System unbrauchbar oder nicht verwaltbar wird, muss es mindestens immer einen Volladministrator geben. Daher kann dieser Administrator nicht deaktiviert oder gelöscht werden. Sie können einen druckbaren HTML-Bericht erstellen, auf dem alle Rollen, Geltungsbereiche und Berechtigungen eines Administrators aufgeführt werden.

Die Volladministratorrolle hat immer den Geltungsbereich "Alle"; dies kann nicht geändert werden.

1. Klicken Sie in Studio auf Konfiguration > Administratoren im linken Bereich und dann auf die Registerkarte Administratoren im mittleren Bereich. Eine Liste der vorhandenen Administratoren wird angezeigt.
2. Klicken Sie im Bereich "Aktionen" auf Administrator erstellen.
3. Geben Sie den Namen des Administratorkontos ein oder durchsuchen Sie die Ordner.
4. Um festzulegen, auf welche Objekte der Administrator zugreifen kann, wählen Sie einen Geltungsbereich aus. Sie können einen der vorhandenen Geltungsbereiche auswählen oder einen neuen erstellen (weitere Informationen finden Sie unter [Erstellen eines Geltungsbereichs](#)).
5. Wählen Sie eine Rolle aus, um zu definieren, welche Berechtigungen der Administrator für die Objekte des Geltungsbereichs hat.
6. Eine Zusammenfassung der neuen Administratordetails wird angezeigt. Standardmäßig ist der neue Administrator aktiviert. Um den neuen Administrator zu deaktivieren, entfernen Sie die Markierung im Kontrollkästchen Administrator aktivieren.
Klicken Sie auf Create a full permissions report for this Administrator (HTML format), um einen Bericht zu erstellen, in dem alle Berechtigungen des Administrators aufgeführt werden.
7. Klicken Sie zum Erstellen des Administrators auf Fertig stellen.

Sie können einen neuen Administrator durch Kopieren der Informationen eines vorhandenen Administrators erstellen. Der neue Administrator verfügt über die gleichen Rollen-/Geltungsbereichspaare wie der kopierte Administrator, es sei denn, sie werden geändert.

1. Klicken Sie in Studio auf Konfiguration > Administratoren im linken Bereich und dann auf die Registerkarte Administratoren im mittleren Bereich. Eine Liste der vorhandenen Administratoren wird angezeigt.
2. Wählen Sie im mittleren Bereich den Administrator aus, den Sie kopieren möchten und klicken Sie anschließend auf Administrator kopieren im Aktionsbereich. Die Rollen-/Geltungsbereichspaare des Administrators, den Sie kopieren,

werden angezeigt.

3. Geben Sie den Namen des Administratorkontos ein oder durchsuchen Sie die Ordner.
4. Sie können die Rollen-/Geltungsbereichspaare bearbeiten oder löschen und neue hinzufügen. Standardmäßig wird der neue Administrator aktiviert. Um den neuen Administrator zu deaktivieren, entfernen Sie die Markierung im Kontrollkästchen Administrator aktivieren.

Klicken Sie auf Vollständigen Bericht über Berechtigungen anzeigen, um einen Bericht zu erstellen, in dem alle Berechtigungen des Administrators aufgeführt werden.

5. Klicken Sie zum Erstellen des Administrators auf Speichern.

1. Klicken Sie in Studio auf Konfiguration > Administratoren im linken Bereich und dann auf die Registerkarte Administratoren im mittleren Bereich. Eine Liste der vorhandenen Administratoren wird angezeigt.
2. Wählen Sie im mittleren Bereich den Administrator aus, den Sie kopieren möchten und klicken Sie anschließend auf Administrator bearbeiten im Aktionsbereich. Die Details dieses Administrators werden angezeigt. Sie können sie wie folgt aktualisieren:
 - Klicken Sie auf Hinzufügen, um ein Rollen-/Geltungsbereichspaar hinzuzufügen. Sie werden genauso wie beim Erstellen eines neuen Administrators durch die Schritte zum Hinzufügen von Geltungsbereichen und Rollen geführt.
 - Klicken Sie auf das gewünschte Paar und dann auf Bearbeiten, um ein Rollen-/Geltungsbereichspaar zu bearbeiten. Es wird eine Liste der Geltungsbereiche angezeigt; der derzeit mit der Rolle gepaarte Geltungsbereich ist markiert. Sie können entweder die Objekte ändern, für die der Administrator berechtigt ist, indem Sie einen anderen Geltungsbereich auswählen, oder Sie können die Berechtigungen des Administrators für die Objekte ändern, indem Sie eine andere Rolle auswählen:
 - Zur Auswahl eines anderen Geltungsbereichs klicken Sie darauf und dann auf Speichern.
 - Zum Erstellen eines neuen Geltungsbereichs klicken Sie auf Neuen Geltungsbereich erstellen (weitere Informationen finden Sie unter [Erstellen eines Geltungsbereichs](#)).
 - Zur Auswahl einer anderen Rolle klicken Sie im linken Bereich auf Rolle. Es wird eine Liste der Rollen angezeigt; die derzeit mit dem Geltungsbereich gepaarte Rolle ist markiert. Klicken Sie auf die neue Rolle und dann auf Speichern.
 - Zum Erstellen einer neuen Rolle klicken Sie auf Neue Rolle erstellen (weitere Informationen finden Sie unter [Erstellen einer Rolle](#)).
 - Klicken Sie auf das gewünschte Paar und dann auf Löschen, um ein Rollen-/Geltungsbereichspaar zu entfernen. Hiermit wird nur die Beziehung zwischen der Rolle und dem Geltungsbereich für diesen Administrator gelöscht; es wird weder die Rolle noch der Geltungsbereich gelöscht und andere Administratoren mit diesem Rollen-/Geltungsbereichspaar sind ebenfalls nicht betroffen.
 - Zum Aktivieren bzw. Deaktivieren des Administrators klicken Sie auf Administrator aktivieren. Der Administrator kann nicht deaktiviert oder entfernt werden, wenn dies dazu führt, dass kein aktivierter Volladministrator mehr vorhanden ist.

Ein Administrator kann nicht gelöscht werden, wenn dies dazu führt, dass kein Volladministrator mehr vorhanden ist.

1. Wählen Sie in Studio im linken Bereich Konfiguration > Administratoren aus. Eine Liste der vorhandenen Administratoren wird angezeigt.
2. Wählen Sie im mittleren Bereich den Administrator aus, den Sie löschen möchten und klicken Sie anschließend auf Administrator löschen. Die diesem Administrator zugeordneten Geltungsbereiche und Rollen werden angezeigt, damit Sie die Auswirkungen des Löschvorgangs überprüfen können, bevor Sie den Löschvorgang bestätigen.

Sie können einen druckbaren HTML-Bericht erstellen, auf dem alle Berechtigungen eines Administrators aufgeführt werden. Der Bericht zeigt die Rollen-/Geltungsbereichspaare, die dem Administrator zugeordnet sind, sowie die einzelnen Berechtigungen für jeden Objekttyp (z. B. Bereitstellungsgruppen und Maschinenkataloge).

1. Wählen Sie in Studio im linken Bereich Konfiguration > Administratoren aus. Eine Liste der vorhandenen Administratoren wird angezeigt.
2. Klicken Sie im mittleren Bereich auf den Administrator und anschließend auf Richtlinienresultatsatz-Bericht erstellen.

Sie können auch beim Erstellen, Kopieren oder Bearbeiten eines Administrators einen Bericht erstellen.

Rollen

Sep 29, 2015

Sie können Informationen über alle Rollen anzeigen. Sie können eine integrierte Rolle kopieren. Sie können eine benutzerdefinierte Rolle erstellen, bearbeiten oder löschen.

Rollenamen können maximal 64 Unicode-Zeichen enthalten. Die folgenden Zeichen sind nicht zulässig: \ (umgekehrter Schrägstrich), / (Schrägstrich), ; (Semikolon), : (Doppelpunkt), # (Nummernzeichen), , (Komma), * (Sternchen), ? (Fragezeichen), = (Gleichheitszeichen), < (Pfeil nach links), > (Pfeil nach rechts), | (senkrechter Strich), [] (eckige Klammern), () (runde Klammern), " (Anführungszeichen) und ' (Apostroph). Beschreibungen können bis zu 256 Unicode-Zeichen enthalten.

Hinweis: Nur bestimmte Produkteditionen unterstützen benutzerdefinierte Rollen. Editionen, die keine benutzerdefinierten Rollen unterstützen, verfügen nicht über verwandte Einträge im Aktionsbereich.

1. Klicken Sie in Studio im linken Bereich auf Konfiguration > Administratoren und klicken Sie anschließend auf die Registerkarte Rollen im mittleren Bereich. Eine Liste der vorhandenen Rollen wird angezeigt.
2. Wählen Sie die Rolle aus, die Sie anzeigen möchten. Im unteren Teil des mittleren Bereichs werden die Objekttypen und die zugehörigen Berechtigungen für die Rolle angezeigt. Klicken Sie auf die Registerkarte Administratoren im unteren Bereich, um eine Liste der Administratoren anzuzeigen, die derzeit diese Rolle haben.

1. Klicken Sie in Studio im linken Bereich auf Konfiguration > Administratoren und klicken Sie anschließend auf die Registerkarte Rollen im mittleren Bereich. Eine Liste der vorhandenen Rollen wird angezeigt.
2. Klicken Sie im Bereich "Aktionen" auf Rolle erstellen.
3. Geben Sie einen Namen und eine Beschreibung für die Rolle ein.
4. Markieren Sie jeden Objekttyp, für den die Rolle Berechtigungen haben soll, und wählen Sie anschließend die Berechtigungen.
5. Klicken Sie auf Save.

Sie können eine Rolle durch Kopieren der Informationen einer vorhandenen Rolle erstellen. Sie können Rollen nur dann kopieren, wenn Sie zum Erstellen benutzerdefinierter Rollen berechtigt sind.

1. Klicken Sie in Studio im linken Bereich auf Konfiguration > Administratoren und klicken Sie anschließend auf die Registerkarte Rollen im mittleren Bereich. Eine Liste der vorhandenen Rollen wird angezeigt.
2. Klicken Sie im mittleren Bereich auf die zu kopierende Rolle und klicken Sie anschließend im Aktionsbereich auf Rolle kopieren.
3. Geben Sie einen Namen und eine Beschreibung für die neue Rolle ein.
4. Bearbeiten Sie die Objekttypen und Berechtigungen der Rolle nach Bedarf.
5. Klicken Sie auf Save.

Bevor Sie eine benutzerdefinierte Rolle bearbeiten, stellen Sie sicher, dass die Änderungen keine negativen Auswirkungen auf die Administratoren mit dieser Rolle haben. Integrierte Rollen können nicht bearbeitet werden.

1. Klicken Sie in Studio im linken Bereich auf Konfiguration > Administratoren und klicken Sie anschließend auf die

Registerkarte Rollen im mittleren Bereich. Eine Liste der vorhandenen Rollen wird angezeigt.

2. Klicken Sie im mittleren Bereich auf die zu bearbeitende Rolle und klicken Sie anschließend im Aktionsbereich auf Rolle bearbeiten.
3. Bearbeiten Sie den Namen und die Beschreibung sowie die Objekttypen und Berechtigungen der Rolle nach Bedarf.
4. Klicken Sie auf Save.

Benutzerdefinierte Rollen können nicht gelöscht werden, wenn sie von einem Administrator verwendet werden. Integrierte Rollen können nicht gelöscht werden.

1. Klicken Sie in Studio im linken Bereich auf Konfiguration > Administratoren und klicken Sie anschließend auf die Registerkarte Rollen im mittleren Bereich. Eine Liste der vorhandenen Rollen wird angezeigt.
2. Klicken Sie im mittleren Bereich auf die zu löschende Rolle und klicken Sie anschließend im Aktionsbereich auf Rolle löschen.
3. Bestätigen Sie die Löschung.

Geltungsbereiche

Sep 29, 2015

Beim Erstellen einer Site steht nur der Geltungsbereich "Alle" zur Verfügung. Dieser kann nicht gelöscht werden.

Sie können Geltungsbereiche mit den Schritten in diesem Abschnitt erstellen. Sie können auch die Geltungsbereiche erstellen, wenn Sie einen Administrator erstellen. Jeder Administrator muss mindestens einem Rollen-/Geltungsbereichspaar zugeordnet werden. Beim Erstellen oder Bearbeiten von Desktops, Maschinenkatalogen, Anwendungen oder Hosts können Sie diese einem bestehenden Geltungsbereich hinzufügen. Wenn Sie sie keinem Geltungsbereich hinzufügen, bleiben sie Teil des Geltungsbereichs "Alle".

Die Geltungsbereichszuordnung ist bei der Erstellung von Sites und für Objekte der delegierten Administration (Geltungsbereiche und Rollen) nicht möglich. Objekte, die nicht zugeordnet werden können, gehören zum Geltungsbereich "Alle". (Volladministratoren ist immer der Geltungsbereich "Alle" zugeordnet.) Maschinen, Energieaktionen, Desktops und Sitzungen werden nicht direkt einem Geltungsbereich zugeordnet. Administratoren können Berechtigungen für diese Objekte über die zugehörigen Maschinenkataloge oder Bereitstellungsgruppen zugewiesen werden.

Geltungsbereichsnamen können maximal 64 Unicode-Zeichen enthalten. Die folgenden Zeichen sind nicht zulässig: \ (umgekehrter Schrägstrich), / (Schrägstrich), ; (Semikolon), : (Doppelpunkt), # (Nummernzeichen), , (Komma), * (Sternchen), ? (Fragezeichen), = (Gleichheitszeichen), < (Pfeil nach links), > (Pfeil nach rechts), | (senkrechter Strich), [] (eckige Klammern), () (runde Klammern), " (Anführungszeichen) und ' (Apostroph). Beschreibungen können bis zu 256 Unicode-Zeichen enthalten.

1. Klicken Sie in Studio im linken Bereich auf Konfiguration > Administratoren und klicken Sie anschließend auf die Registerkarte Geltungsbereiche im mittleren Bereich. Eine Liste der vorhandenen Geltungsbereiche wird angezeigt.
2. Klicken Sie im Bereich "Aktionen" auf Geltungsbereich erstellen.
3. Geben Sie einen Namen (z. B. "Vertrieb") und eine Beschreibung (z. B. "Bereitstellungsgruppen der Vertriebsabteilung") für den Geltungsbereich an.
4. Wählen Sie die Objekttypen oder bestimmte Objekte:
 1. Zum Wählen aller Objekte eines bestimmten Typs (z. B. Bereitstellungsgruppen), wählen Sie den Objekttyp.
 2. Zum Wählen bestimmter Objekte erweitern Sie den Typ und wählen Sie die einzelnen Objekte (z. B. einzelne Bereitstellungsgruppen des Vertriebs) aus.
5. Klicken Sie auf Speichern.

Sie können einen Geltungsbereich durch Kopieren eines vorhandenen Geltungsbereichs erstellen.

1. Klicken Sie in Studio im linken Bereich auf Konfiguration > Administratoren und klicken Sie anschließend auf die Registerkarte Geltungsbereiche im mittleren Bereich. Eine Liste der vorhandenen Geltungsbereiche wird angezeigt.
2. Klicken Sie im mittleren Bereich auf den zu kopierenden Geltungsbereich und klicken Sie anschließend im Aktionsbereich auf Geltungsbereich kopieren.
3. Geben Sie einen Namen und eine Beschreibung für den neuen Geltungsbereich ein.
4. Bearbeiten Sie die Objekttypen und Berechtigungen des Geltungsbereichs nach Bedarf.
5. Klicken Sie auf Speichern.

1. Wählen Sie in Studio im linken Bereich Konfiguration > Administratoren aus. Klicken Sie dann im mittleren Bereich auf die Registerkarte Geltungsbereiche. Eine Liste der vorhandenen Geltungsbereiche wird angezeigt.
2. Klicken Sie im mittleren Bereich auf den zu bearbeitenden Geltungsbereich und klicken Sie anschließend im Aktionsbereich auf Geltungsbereich bearbeiten.
3. Bearbeiten Sie den Namen und die Beschreibung sowie die Objekttypen und Berechtigungen des Bereichs nach Bedarf. Wichtig: Werden Objekten aus einem Geltungsbereich entfernt, kann der Administrator, der sie entfernt hat, möglicherweise nicht mehr auf sie zugreifen. Ist der bearbeitete Geltungsbereich mit einer oder mehreren Rollen gepaart, stellen Sie sicher, dass durch Änderungen an dem Bereich kein Rollen-/Geltungsbereichspaar unbrauchbar wird.
4. Klicken Sie auf Speichern.

Geltungsbereiche können nicht gelöscht werden, wenn sie von einem Administrator verwendet werden.

1. Klicken Sie in Studio im linken Bereich auf Konfiguration > Administratoren und klicken Sie anschließend im mittleren Bereich auf die Registerkarte Geltungsbereiche. Eine Liste der vorhandenen Geltungsbereiche wird angezeigt.
2. Klicken Sie im mittleren Bereich auf den zu löschenden Geltungsbereich und klicken Sie anschließend im Aktionsbereich auf Geltungsbereich löschen.
3. Bestätigen Sie die Löschung.

Berichte

Sep 29, 2015

Sie können folgende Berichte für die delegierte Administration erstellen:

- HTML-Bericht mit den Berechtigungen für einen einzelnen Administrator. Dies ist ein Richtlinienergebnissatz-Bericht, der über Citrix Studio angefordert wird.
- HTML- oder CSV Bericht, in dem alle integrierten benutzerdefinierten Rollen und Berechtigungen zugeordnet sind. Dieser Bericht wird unter Ausführen eines PowerShell-Skripts erstellt.

Dieser HTML-Bericht enthält die Rollen-/Geltungsbereichspaare, die einem Administrator zugeordnet sind, sowie eine Liste der einzelnen Berechtigungen für jeden Objekttyp (z. B. Bereitstellungsgruppen, und Maschinenkataloge).

1. Wählen Sie in Studio im linken Bereich Konfiguration > Administratoren aus. Eine Liste der vorhandenen Administratoren wird angezeigt.
2. Wählen Sie im mittleren Bereich einen Administrator aus, und klicken Sie dann auf Richtlinienergebnissatz-Bericht erstellen.

Sie können diesen Bericht auch beim Erstellen, Kopieren oder Bearbeiten eines Administrators anfordern.

Verwenden Sie das PowerShell-Skript OutputPermissionMapping.ps1 zum Erstellen einer HTML- oder CSV-Tabelle mit den Berechtigungen, die jeder integrierten und benutzerdefinierten Rolle zugeordnet sind.

Um dieses Skript auszuführen, müssen Sie ein Volladministrator, ein Lesezugriffadministrator oder ein benutzerdefinierter Administrator mit der Berechtigung zum Lesen von Rollen sein. Das Skript befindet sich in: Programme\Citrix\DelegatedAdmin\SnapIn\Citrix.DelegatedAdmin.Admin.V1\Scripts\.

Syntax:

OutputPermissionMapping.ps1 [-Help] [-Csv] [-Path <string>] [-AdminAddress <string>] [-Show] [<CommonParameters>]

Parameter	Beschreibung
-Help	Zeigt Skripthilfe an.
-Csv	Gibt CSV-Ausgabe an. Standard = HTML
-Path	Gibt an, wo die Ausgabe gespeichert werden soll. Standard = stdout
-AdminAddress	Gibt die IP-Adresse oder den Hostnamen des Delivery Controllers an, mit dem eine Verbindung hergestellt wird. Standard = localhost
-Show	Gilt nur, wenn der Parameter "-Path" ebenfalls angegeben wird. Wenn die Ausgabe in eine Datei geschrieben wird, wird sie mit diesem Parameter einem geeigneten Programm, z. B. einem Webbrowser, geöffnet.
<Allgemeine Parameter>	Allgemeine Parameter, die von diesem Skript unterstützt werden sind: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer und OutVariable. Weitere Informationen finden Sie in der Dokumentation von Microsoft.

Mit dem Befehl im folgenden Beispiel wird eine HTML-Tabelle in eine Datei namens Roles.html geschrieben und die Tabelle in einem Webbrowser geöffnet.

& "\$env:ProgramFiles\Citrix\DelegatedAdmin\SnapIn\Citrix.DelegatedAdmin.Admin.V1\Scripts\OutputPermissionMapping.ps1" -Path Roles.html -Show

Mit dem Befehl im folgenden Beispiel wird eine CSV-Tabelle in eine Datei namens Roles.csv geschrieben. Die Tabelle wird nicht angezeigt.

& "\$env:ProgramFiles\Citrix\DelegatedAdmin\SnapIn\Citrix.DelegatedAdmin.Admin.V1\Scripts\OutputPermissionMapping.ps1" -CSV -Path Roles.csv

An einer Windows-Eingabeaufforderung wird der Befehl aus dem vorherigen Beispiel folgendermaßen eingegeben:

powershell -command "& '%ProgramFiles%\Citrix\DelegatedAdmin\SnapIn\Citrix.DelegatedAdmin.Admin.V1\Scripts\OutputPermissionMapping.ps1' -CSV -Path Roles.csv"

IPv6

Sep 29, 2015

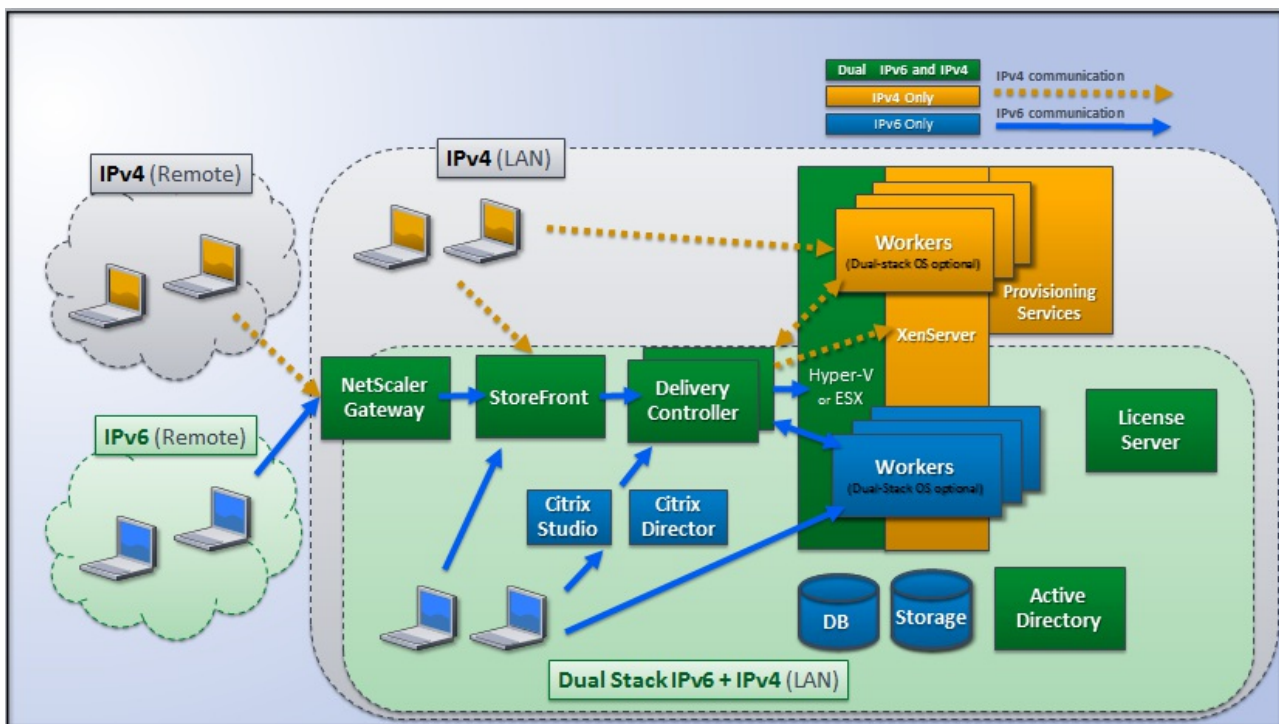
Dieses Release unterstützt reines IPv4, reines IPv6 und Bereitstellungen mit dualtem Stapel, bei denen überlappende IPv4- und IPv6-Netzwerke verwendet werden.

Die IPv6-Kommunikation wird mit zwei verbindungspezifischen Citrix Richtlinieneinstellungen für Virtual Delivery Agent (VDA) gesteuert:

- Die primäre Einstellung erzwingt die Verwendung von IPv6: Nur IPv6-Controllerregistrierung verwenden.
- Die abhängige Einstellung definiert eine IPv6-Netzwerkmaske: IPv6-Netzwerkmaske für Controllerregistrierung.

Wenn Nur IPv6-Controllerregistrierung verwenden aktiviert ist, erfolgt die VDA-Registrierung bei einem Delivery Controller für eingehende Verbindungen über eine IPv6-Adresse.

Die folgende Abbildung zeigt eine IPv4-/IPv6-Bereitstellung mit dualtem Stapel. In diesem Szenario ist ein Worker ein auf einem Hypervisor oder auf einer physischen Maschine installierter VDA, der primär zum Aktivieren von Verbindungen für Anwendungen und Desktops verwendet wird. Komponenten, die für den Parallelbetrieb von IPv6 und IPv4 ausgelegt sind, werden auf Betriebssystemen ausgeführt, die Tunneling oder Dual Protocol-Software nutzen.



Diese Citrix Produkte, Komponenten und Features unterstützen nur IPv4:

- Provisioning Services
- XenServer Version 6.x
- VDAs, die nicht mit der Richtlinieneinstellung Nur IPv6-Controllerregistrierung verwenden gesteuert werden
- XenApp-Versionen vor 7.5, XenDesktop-Versionen vor 7 und EdgeSight

In dieser Bereitstellung gilt:

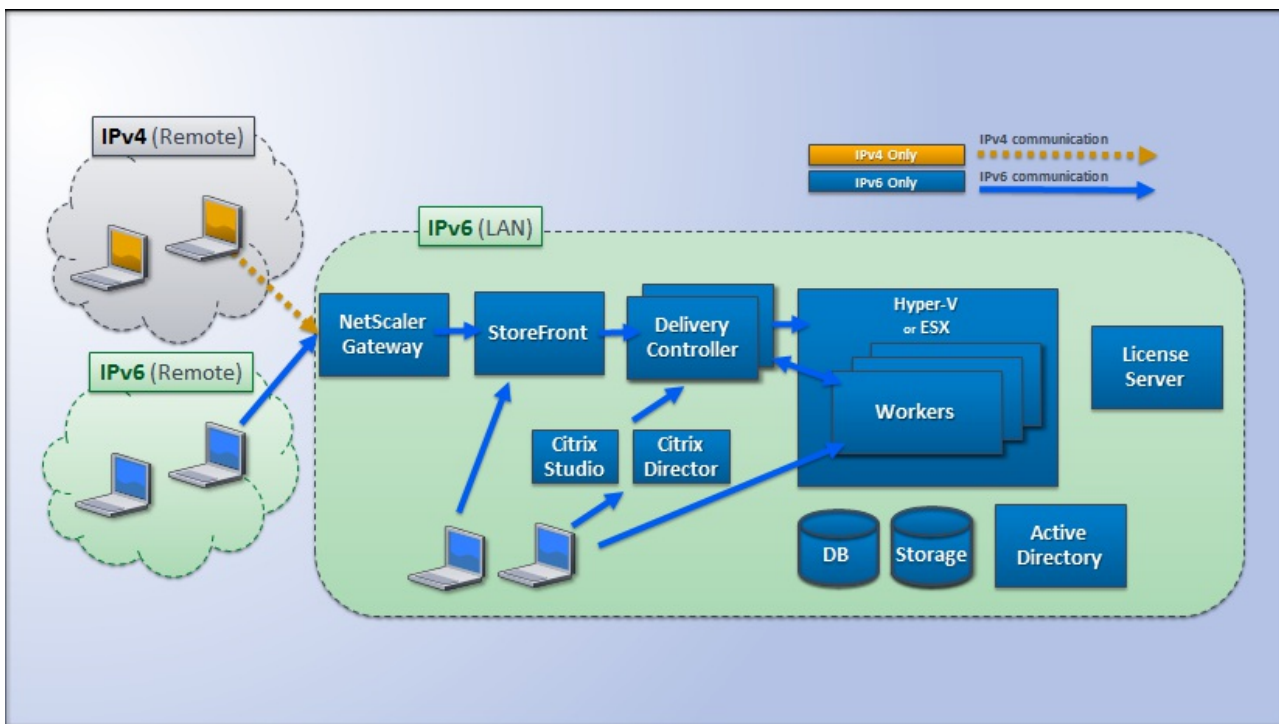
- Wenn ein Team häufig ein IPv6-Netzwerk verwendet und der Administrator die Nutzung von IPv6-Datenverkehr

wünscht, veröffentlicht der Administrator IPv6-Desktops und -Anwendungen für die Benutzer auf der Basis eines Workerimages oder von Organisationseinheiten (OU), für das bzw. die die primäre IPv6-Richtlinieneinstellung (Nur IPv6-Controllerregistrierung verwenden) aktiviert ist.

- Wenn ein Team häufig ein IPv4-Netzwerk verwendet, veröffentlicht der Administrator IPv4-Desktops und -Anwendungen für die Benutzer auf der Basis eines Workerimages oder von Organisationseinheiten (OU), für das bzw. die die primäre IPv6-Richtlinieneinstellung deaktiviert ist (d. h. Nur IPv6-Controllerregistrierung verwenden ist deaktiviert = Standardeinstellung).

Die folgende Abbildung zeigt eine reine IPv6-Bereitstellung. Für dieses Szenario gilt:

- Die Komponenten werden auf Betriebssystemen ausgeführt, die ein IPv6-Netzwerk unterstützen.
- Die primäre Citrix Richtlinieneinstellung (Nur IPv6-Controllerregistrierung verwenden) ist für alle VDAs aktiviert; sie müssen sich beim Controller mit einer IPv6-Adresse registrieren.



Wenn Ihre Umgebung sowohl IPv4- und IPv6-Netzwerke umfasst, benötigen Sie separate Bereitstellungsgruppenkonfigurationen für IPv4-exklusive Clients und für die Clients, die Zugriff auf das IPv6-Netzwerk haben. Verwenden Sie ggf. Namen, die manuelle Active Directory-Gruppenzuweisung oder SmartAccess-Filter zur Unterscheidung der Benutzer.

Die Wiederverbindung mit einer Sitzung kann fehlschlagen, wenn die Verbindung auf einem IPv6-Netzwerk initiiert wird und dann Wiederverbindungsversuche von einem internen Client erfolgen, der nur über IPv4-Zugriff verfügt.

Bei reiner IPv6-Implementierung bzw. Implementierung von IPv4/IPv6 mit dualem Stapel sind zwei Citrix Richtlinieneinstellungen relevant. Konfigurieren Sie die folgenden verbindungsbezogenen Einstellungen:

- **Nur IPv6 Controllerregistrierung verwenden:** Steuert das Adressformat, mit dem der Virtual Delivery Agent (VDA) beim Delivery Controller registriert wird. Standard = deaktiviert

- Wenn der VDA mit dem Controller kommuniziert, wird eine IPv6-Adresse verwendet, deren Auswahl folgender Reihenfolge unterliegt: globale IP-Adresse, Unique Local Address (ULA), Link-Local-Adresse (nur wenn keine anderen IPv6-Adressen verfügbar sind).
- Bei deaktivierter Einstellung wird der VDA mit der IPv4-Adresse der Maschine für die Kommunikation beim Controller registriert.
- **IPv6-Netzwerkmaske für Controllerregistrierung:** Eine Maschine kann über mehrere IPv6-Adressen verfügen. Mit dieser Richtlinieneinstellung können Administratoren den VDA auf ein bevorzugtes Subnetz beschränken (anstelle einer globalen IP, sofern registriert). Mit dieser Einstellung wird das Netzwerk festgelegt, in dem der VDA registriert wird: Der VDA wird nur bei der ersten Adresse registriert, die mit der angegebenen Netzwerkmaske übereinstimmt. Diese Einstellung ist nur gültig, wenn die Richtlinieneinstellung Nur IPv6-Controllerregistrierung verwenden aktiviert ist. Standard = leere Zeichenfolge

Wichtig: Die Verwendung von IPv4 oder IPv6 durch einen VDA wird ausschließlich über diese Richtlinieneinstellungen gesteuert. Das bedeutet, um die IPv6-Adressierung nutzen zu können, muss der VDA von einer Citrix Richtlinie gesteuert werden, bei der die Einstellung Nur IPv6-Controllerregistrierung verwenden aktiviert ist.

Konfigurieren der Zeitzoneneinstellungen

Sep 29, 2015

Wenn Benutzer ohne Superuserrechte eine Verbindung mit Windows XP-Desktops herstellen, wird ihnen standardmäßig die Zeitzone des Systems angezeigt, auf dem der Desktop ausgeführt wird, nicht aber die Zeitzone ihres eigenen Benutzergeräts. Dies gilt nicht für Windows Vista oder Windows 7, die über eine separate Zeitzoneberechtigung verfügen.

Damit Windows XP-Benutzer die lokale Uhrzeit anzeigen können, müssen Sie ihnen folgende Rechte zuweisen:

- Ändern Sie die Uhrzeit Zeit auf dem System, auf dem der Desktop ausgeführt wird. Richten Sie dafür eine Gruppenrichtlinie ein, mit der Sie nicht-berechtigten Benutzern das Recht geben, die Systemzeiteinstellungen zu ändern. Weitere Informationen hierzu finden Sie unter <http://msdn2.microsoft.com/en-us/library/ms813808.aspx>.
- Ändern Sie den Zeitzoneregistrierungsbereich. Weitere Informationen finden Sie unter <http://support.microsoft.com/kb/300022/>.

Anschließend wird Benutzern, die Verbindungen mit Windows XP-Desktops herstellen, die lokale Zeitzone auf dem Desktop angezeigt. Wenn sie sich abmelden oder die Sitzung trennen, wird die Zeitzone auf dem Desktop wieder auf die Einstellung zurückgesetzt, die verwendet wurde, bevor sie sich angemeldet haben.

Sie können Zeitzoneneinstellungen über Citrix Richtlinien konfigurieren. Verwenden Sie die Richtlinieneinstellung Lokale Zeit des Clients verwenden im Abschnitt "Zeitzonesteuerung" des Ordners mit den ICA-Richtlinieneinstellungen.

Konfigurieren von Verbindungstimern

Sep 29, 2015

Sie können drei Verbindungstimer konfigurieren:

- Timer für längste Verbindung: Diese Einstellung legt die Höchstdauer einer ununterbrochenen Verbindung zwischen einem Benutzergerät und einem Desktop fest. Verwenden Sie die Richtlinieneinstellungen "Sitzungsverbindungstimer" und "Sitzungsverbindung - Timerintervall", um dies zu konfigurieren.
- Timer für inaktive Verbindungen: Diese Einstellung legt fest, wie lange eine ununterbrochene Verbindung zwischen einem Benutzergerät und einem virtuellen Desktop erhalten wird, wenn keine Eingabe vom Benutzer erfolgt. Verwenden Sie die Richtlinieneinstellungen "Sitzungsleerlaufstimer" und "Sitzungsleerlauf - Timerintervall", um dies zu konfigurieren.
- Timer für getrennte Verbindungen: Diese Einstellung legt fest, wie lange ein getrennter, gesperrter virtueller Desktop gesperrt bleibt, bis die Sitzung abgemeldet wird. Verwenden Sie die Richtlinieneinstellungen "Timer für getrennte Sitzung" und "Getrennte Sitzungen - Timerintervall", um dies zu konfigurieren.

Wenn Sie diese Einstellungen aktualisieren müssen, achten Sie darauf, dass sie in der ganzen Bereitstellung konsistent sind.

Konfigurieren der USB-Unterstützung

Sep 29, 2015

Mit HDX USB-Geräteumleitung können USB-Geräte zum und vom Benutzergerät umgeleitet werden. Benutzer können beispielsweise eine Verbindung zum USB-Flashlaufwerk auf einem lokalen Computer herstellen und dann remote von einem virtuellen Desktop oder einer desktopgehosteten Anwendung darauf zugreifen. Während einer Sitzung können Benutzer die Geräte mit Plug & Play verwenden, einschließlich PTP-Geräte (Picture Transfer Protocol) wie digitale Kameras, MTP-Geräte (Media Transfer Protocol) wie digitale Audio Player oder tragbare Media Player und POS-Geräte (Point of Sale).

Double-Hop-USB wird bei Sitzungen mit über Desktop gehosteten Anwendungen nicht unterstützt.

USB-Umleitung ist für folgende Receiver verfügbar:

- Receiver für Windows
- Receiver für Linux

Standardmäßig ist die USB-Umleitung für bestimmte Klassen von USB-Geräten zulässig bzw. nicht zulässig. Weitere Informationen finden Sie in der Dokumentation für Receiver. Sie können festlegen, welche USB-Geräte einem virtuellen Desktop verfügbar gemacht werden, indem Sie die Liste der unterstützten USB-Geräte für die Umleitung beschränken. Weiteres hierzu finden Sie weiter unten in diesem Thema.

Wichtig: In Umgebungen, in denen aus Sicherheitsgründen eine Trennung zwischen dem Benutzergerät und dem Server erforderlich ist, müssen Sie die Benutzer darüber informieren, welche Typen von USB-Geräten zu vermeiden sind.

Die beliebtesten USB-Geräte können über optimierte virtuelle Kanäle umgeleitet werden und liefern hervorragende Leistung und Bandbreiteneffizienz über ein WAN. Der Grad der Unterstützung hängt vom auf dem Benutzergerät installierten Receiver ab; Informationen hierzu finden Sie in der Dokumentation für Receiver. Optimierte virtuelle Kanäle sind normalerweise die beste Option, insbesondere in Umgebungen mit hoher Latenz.

Hinweis: Für die Umleitung von USB-Geräten gilt, dass ein SMART-Board ebenso wie eine Maus behandelt wird.

Es gibt keine Unterstützung für die generische USB-Umleitung von Geräten, die mit USB 3.0-Ports verbunden sind. Das Produkt unterstützt die Verwendung von optimierten virtuellen Kanälen mit USB 3.0-Geräten und USB 3.0-Ports, wie ein virtueller CDM-Kanal, der zum Anzeigen von Dateien auf einer Kamera oder zur Audiowiedergabe über ein Headset verwendet wird. Das Produkt unterstützt zudem die generische USB-Umleitung von USB 3.0-Geräten, die mit einem USB 2.0-Port verbunden sind.

Spezialgeräte, für die es keine optimierten virtuellen Kanäle gibt, werden durch die Verwendung eines allgemeinen virtuellen Kanals für USB unterstützt, der eine unformatierte USB-Umleitung bietet. Informationen zu allen mit XenDesktop getesteten USB-Geräten finden Sie unter <http://support.citrix.com/article/ctx123569>.

Generische USB-Umleitung wird auf Desktopbetriebssystem-VDA, jedoch nicht auf Serverbetriebssystem-VDA unterstützt.

Einige erweiterte gerätespezifischen Features, wie z. B. auf HID-Schaltflächen (Human Interface Device) auf einer Webcam, funktionieren möglicherweise mit dem optimierten virtuellen Kanal nicht wie erwartet. Wenn dies ein Problem darstellt, können Sie den allgemeinen virtuellen Kanal für USB verwenden.

Einige Geräte werden standardmäßig nicht umgeleitet und sind nur in der lokalen Sitzung verfügbar. Es wäre beispielsweise nicht angemessen, eine über internes USB an der Hauptplatine des Benutzergeräts angeschlossene Netzwerkkarte umzuleiten.

Die folgenden Citrix Richtlinieninstellungen steuern die USB-Unterstützung:

- Client-USB-Geräteumleitung: Standardeinstellung: Nicht zugelassen.
- Regeln für die Client-USB-Geräteumleitung: Die Regeln gelten nur für Geräte, die generische USB-Umleitung verwenden. Die Regeln gelten daher nicht für Geräte, die spezielle oder optimierte Umleitung verwenden, wie CDM.
- Client-USB-Geräteumleitung für Plug & Play-Geräte: Die Standardeinstellung Zugelassen ermöglicht Plug & Play von PTP-, MTP- und POS-Geräten in einer Sitzung.
- Bandbreitenlimit für Client-USB-Geräteumleitung: Der Standardwert ist 0 (d. h. ohne Maximum).
- Bandbreitenlimit für Client-USB-Geräteumleitung (Prozent): Der Standardwert ist 0 (d. h. ohne Maximum).

1. Fügen Sie die Einstellung Client-USB-Geräteumleitung einer Richtlinie hinzu und stellen Sie den Wert auf Zugelassen ein.
2. (Optional) Zum Aktualisieren der für Remoting verfügbaren USB-Geräteliste fügen Sie die Einstellung Regeln für die Client-USB-Geräteumleitung einer Richtlinie hinzu und stellen Sie die USB-Standardrichtlinienregeln wie weiter unten in diesem Thema beschrieben ein.
3. Aktivieren Sie die USB-Unterstützung, wenn Sie Receiver auf Benutzergeräten installieren. Weitere Informationen zum Konfigurieren finden Sie in der Receiver-Dokumentation in den eDocs. Wenn Sie im vorigen Schritt die USB-Richtlinienregeln für Virtual Delivery Agent festgelegt haben, geben Sie nun die gleichen Richtlinienregeln für Receiver ein.
Hinweis: Informationen zur USB-Unterstützung Für Thin Clients und die erforderliche Konfiguration erhalten Sie vom Hersteller.

USB-Geräte werden automatisch umgeleitet, wenn die USB-Unterstützung aktiviert ist und die USB-Einstellungen für eine automatische Verbindung der USB-Geräte konfiguriert wurden. USB-Geräte werden auch automatisch umgeleitet, wenn das Gerät im Modus "Desktop Appliance" ist und der Verbindungsbalken nicht angezeigt wird. In manchen Fällen empfiehlt es sich jedoch, die automatische Umleitung nicht für alle USB-Geräte zu aktivieren. Weitere Informationen finden Sie unter <CTX123015>.

Benutzer können Geräte, die nicht automatisch umgeleitet werden, explizit umleiten, indem sie sie aus der USB-Geräteliste auswählen. Um zu verhindern, dass Geräte überhaupt aufgeführt oder umgeleitet werden, können Sie auf dem Client und für VDA die nachfolgenden Geräteregeln festlegen.

Sie können die USB-Geräte, die für Remoting verfügbar sind, aktualisieren, indem Sie Regeln für die Client-USB-Geräteumleitung für Receiver und Virtual Desktop Agent definieren, die dann die USB-Standardrichtlinienregeln überschreiben.

- Bearbeiten Sie die Benutzergeräteregistrierung. Eine administrative Vorlage (ADM-Datei) ist auf dem Installationsmedium enthalten, sodass Sie das Gerät über die Active Directory-Gruppenrichtlinie ändern können: `dvd root \os\lang\Support\Configuration\iclient_usb.adm`.
- Bearbeiten Sie die Administrator-Override-Regeln der VDA-Registrierung auf den Serverbetriebssystemmaschinen. Eine administrative Vorlage (ADM-Datei) ist auf dem Installationsmedium enthalten, sodass Sie VDA über die Active Directory-Gruppenrichtlinie ändern können: `dvd root \os\lang\Support\Configuration\vda_usb.adm`.

Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab.

Die Produktstandardregeln sind in HKLM\SOFTWARE\Citrix\PortICA\GenericUSB\DeviceRules gespeichert. Ändern Sie diese Produktstandardregeln nicht. Verwenden Sie sie als Anleitung zum Erstellen von Administrator-Überschreibungsregeln, wie nachfolgend beschrieben. Die GPO-Überschreibungen werden ausgewertet, bevor die Produktstandardregeln angewendet werden.

Die Administrator-Override-Regeln sind in HKLM\SOFTWARE\Citrix\PortICA\GenericUSB\DeviceRules gespeichert. GPO-Richtlinienregeln haben das Format {Allow:|Deny:} gefolgt von Tag=Wert-Ausdrücken, die durch Leerzeichen getrennt sind. Die folgenden Tags werden unterstützt:

Tag	Beschreibung
VID	Vendor-ID vom Gerätedeskriptor
PID	Produkt-ID vom Gerätedeskriptor
REL	Release-ID vom Gerätedeskriptor
Class	Klasse vom Gerätedeskriptor oder einem Schnittstellendeskriptor; verfügbare USB-Klassencodes finden Sie auf der USB-Website unter http://www.usb.org/ .
SubClass	Unterklasse vom Gerätedeskriptor oder ein Schnittstellendeskriptor
Prot	Protokoll vom Gerätedeskriptor oder ein Schnittstellendeskriptor

- Wenn Sie neue Richtlinienregeln erstellen, beachten Sie Folgendes:
- Bei Regeln wird die Groß- und Kleinschreibung nicht berücksichtigt.
 - Regeln können optional von einem Kommentar gefolgt werden, der mit # eingeleitet wird. Ein Trennzeichen ist nicht erforderlich, der Kommentar wird beim Abgleichen ignoriert.
 - Leere Zeilen und Kommentare werden ignoriert.
 - Leerzeichen dienen als Trennzeichen, sie können nicht in einer Zahl oder Kennung verwendet werden. Beispielsweise ist Deny: Class = 08 SubClass=05 eine gültige Regel; Deny: Class=0 Sub Class=05 hingegen nicht.
 - Tags müssen den Übereinstimmungsoperator = verwenden. Beispielsweise VID=1230.
 - Jede Regel muss auf einer neuen Zeile beginnen oder Teil einer durch Semikolon getrennten Liste sein.
- Wichtig: Wenn Sie die ADM-Vorlagendatei verwenden, müssen Sie die Regeln in einer einzigen Zeile mit Semikolons getrennt eingeben.

Bei optimierten Geräten, wie Massenspeichergeräten, wird das Gerät normalerweise über den speziellen CDM-Kanal umgeleitet und nicht mit Richtlinienregeln. Wenn jedoch eine der folgenden Bedingungen gegeben ist, ist das optimierte Gerät in der Geräteliste im Desktop Viewer für generische USB-Umleitung verfügbar:

- Die automatische Umleitung ist für das Speichergerät festgelegt (z. B. AutoRedirectStorage = 1). Weitere Informationen finden Sie unter [CTX123015](#).
- Geräteverbindungen vereinfachen ist nicht aktiviert. Weitere Informationen finden Sie unter [CTX136716](#).

- Beispiele:
- Das folgende Beispiel zeigt eine vom Administrator definierte USB-Richtlinienregel für Vendor- und Produkt-IDs:
Allow: VID=046D PID=C626 # Allow Logitech SpaceNavigator 3D Mouse Deny: VID=046D # Deny all Logitech products
 - Das folgende Beispiel zeigt eine vom Administrator definierte USB-Richtlinienregel für eine definierte Klasse, Unterklasse und ein Protokoll:
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

Standardmäßig werden USB-Geräte automatisch umgeleitet, wenn die USB-Unterstützung aktiviert ist und die USB-Einstellungen für eine automatische Verbindung der USB-Geräte konfiguriert wurden. USB-Geräte werden auch für Desktopgerätesites oder desktopgehostete Anwendungen automatisch umgeleitet. In manchen Fällen empfiehlt es sich jedoch, die automatische Umleitung nicht für alle USB-Geräte zu aktivieren. Weitere Informationen finden Sie unter <http://support.citrix.com/article/CTX123015>

Benutzer von Desktop Viewer können Geräte umleiten, die nicht automatisch umgeleitet werden, indem Sie sie aus der USB-Geräteliste auswählen. Um zu verhindern, dass Geräte aufgeführt oder umgeleitet werden, können Sie auf dem Benutzergerät und für Virtual Desktop Agent die nachfolgenden Geräteregele festlegen.

- Sie können die USB-Geräte, die für Remoting verfügbar sind, aktualisieren, indem Sie Regeln für die Client-USB-Geräteumleitung für Receiver und Virtual Desktop Agent definieren, die dann die USB-Standardrichtlinienregeln überschreiben. Geräteregele werden sowohl für Receiver als auch Virtual Desktop Agent zwingend umgesetzt. Achten Sie darauf, die Regeln für beide zu ändern, damit die Geräteumleitung wie vorgesehen funktioniert.
- Bearbeiten Sie die Benutzergeräteregistrierung (oder die INI-Dateien bei Receiver für Linux). Weitere Informationen hierzu finden Sie in der Receiver-Dokumentation in den eDocs. Eine administrative Vorlage (ADM-Datei) ist auf dem Installationsmedium enthalten, sodass Sie das Gerät über die Active Directory-Gruppenrichtlinie ändern können: dvd root \os\lang\Support\Configuration\icaclient_usb.adm
 - Bearbeiten Sie die Administrator-Override-Regeln der VDA-Registrierung auf den Serverbetriebssystemmaschinen. Informationen hierzu finden Sie im Folgenden. Eine administrative Vorlage (ADM-Datei) ist auf dem Installationsmedium enthalten, sodass Sie VDA über die Active Directory-Gruppenrichtlinie ändern können: dvd root \os\lang\Support\Configuration\vda_usb.adm

Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab.

Die Produktstandardregeln sind in HKLM\SOFTWARE\Citrix\PortICA\GenericUSB\DeviceRules gespeichert.

Hinweis: Ändern Sie nicht die Produktstandardregeln. Verwenden Sie sie als Anleitung zum Erstellen von Administrator-Überschreibungsregeln, wie nachfolgend beschrieben. Die GPO-Überschreibungen werden ausgewertet, bevor die Produktstandardregeln angewendet werden.

Die Administrator-Überschreibungsregeln sind in HKLM\SOFTWARE\Citrix\PortICA\GenericUSB\DeviceRules gespeichert.

GPO-Richtlinienregeln haben das Format {Allow:|Deny:} gefolgt von Tag=Wert-Ausdrücken, die durch Leerzeichen getrennt sind. Die folgenden Tags werden unterstützt:

Tag	Beschreibung
VID	Vendor-ID vom Gerätedeskriptor
PID	Produkt-ID vom Gerätedeskriptor
REL	Release-ID vom Gerätedeskriptor
Class	Klasse vom Gerätedeskriptor oder einem Schnittstellendeskriptor; verfügbare USB-Klassencodes finden Sie auf der USB-Website unter http://www.usb.org/ .
SubClass	Unterklasse vom Gerätedeskriptor oder ein Schnittstellendeskriptor
Prot	Protokoll vom Gerätedeskriptor oder ein Schnittstellendeskriptor

Wenn Sie neue Richtlinienregeln erstellen, beachten Sie Folgendes:

- Bei Regeln wird die Groß- und Kleinschreibung nicht berücksichtigt.
- Regeln können optional von einem Kommentar gefolgt werden, der mit # eingeleitet wird. Ein Trennzeichen ist nicht erforderlich, der Kommentar wird beim Abgleichen ignoriert.
- Leere Zeilen und Kommentare werden ignoriert.
- Leerzeichen dienen als Trennzeichen, sie können nicht in einer Zahl oder Kennung verwendet werden. Beispielsweise ist Deny: Class = 08 SubClass=05 eine gültige Regel; Deny: Class=0 Sub Class=05 hingegen nicht.
- Tags müssen den Übereinstimmungsoperator = verwenden. Beispielsweise VID=1230.
- Jede Regel muss auf einer neuen Zeile beginnen oder Teil einer durch Semikolon getrennten Liste sein.
Wichtig: Wenn Sie die ADM-Vorlagendatei verwenden, müssen Sie die Regeln in einer einzigen Zeile mit Semikolons getrennt eingeben.

Verwenden der Regeln mit optimierten Geräten

Bei optimierten Geräten, wie Massenspeichergeräten, wird das Gerät normalerweise über den speziellen CDM-Kanal umgeleitet und nicht mit Richtlinienregeln. Wenn jedoch eine der folgenden Bedingungen gegeben ist, ist das optimierte Gerät in der Geräteliste im Desktop Viewer für generische USB-Umleitung verfügbar:

- Die automatische Umleitung ist für das Speichergerät festgelegt (z. B. AutoRedirectStorage = 1).
Weitere Informationen zu dieser Einstellung finden Sie unter [How to Configure Automatic Redirection of USB Devices](#).
- Geräteverbindungen vereinfachen ist nicht aktiviert.
Weitere Informationen zu dieser Einstellung finden Sie unter [USB-Gerät wird nicht im virtuellen Desktop angezeigt](#).

Beispiele

Dieses Beispiel zeigt eine vom Administrator definierte USB-Richtlinienregel für Vendor- und Produkt-IDs:

Allow: VID=046D PID=C626 # Allow Logitech SpaceNavigator 3D Mouse Deny: VID=046D # Deny all Logitech products

Dieses Beispiel zeigt eine vom Administrator definierte USB-Richtlinienregel für eine definierte Klasse, Unterklasse und ein Protokoll:

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

Benutzer können ein USB-Gerät vor oder nach dem Starten einer virtuellen Sitzung anschließen.

Wenn Sie mit Receiver für Windows arbeiten, gilt Folgendes:

- Geräte, die nach dem Sitzungsbeginn angeschlossen werden, erscheinen unmittelbar im USB-Menü von Desktop Viewer.
- Wenn ein USB-Gerät nicht richtig umgeleitet wird, kann das Problem u. U. dadurch behoben werden, dass das Gerät erst nach dem Beginn der virtuellen Sitzung angeschlossen wird.
- Um Datenverlust zu verhindern, verwenden Sie das Windows-Menü Sicheres Entfernen, bevor Sie das USB-Gerät entfernen.

Nur bei Massenspeichergeräten ist Remotezugriff auch durch Clientlaufwerkszuordnung verfügbar. Hierbei werden Laufwerke auf dem Benutzergerät automatisch Laufwerksbuchstaben auf dem virtuellen Desktop zugeordnet, wenn Benutzer sich anmelden. Die Laufwerke werden als freigegebene Ordner mit zugeordneten Laufwerksbuchstaben angezeigt. Um Clientlaufwerkszuordnung zu konfigurieren, verwenden Sie unter den ICA-Richtlinieneinstellungen im Abschnitt "Dateiumleitung" die Einstellung Clientwechseldatenträger.

Die Hauptunterschiede zwischen den beiden Typen der Remotingrichtlinie sind:

Feature	Clientlaufwerkszuordnung	Generische USB-Umleitung
Diese Option ist in der Standardeinstellung aktiviert.	Ja	Nein
Konfigurierbare Leserechte	Ja	Nein
Sicheres Entfernen des Geräts in einer Sitzung	Nein	Ja, unter der Voraussetzung, dass Benutzer den Empfehlungen des Betriebssystems zum sicheren Entfernen von Geräten folgen.

Wenn die Richtlinien für die generische USB-Umleitung und die Clientlaufwerkszuordnung aktiviert sind und ein Massenspeichergerät vor oder nach dem Sitzungsstart angeschlossen wird, wird es mit der Clientlaufwerkszuordnung umgeleitet. Wenn die Richtlinien für die generische USB-Umleitung und die Clientlaufwerkszuordnung aktiviert sind und für ein Gerät die

automatische Umleitung konfiguriert wurde (siehe <http://support.citrix.com/article/CTX123015>) und wenn ein Massenspeichergerät vor oder nach dem Sitzungsstart angeschlossen wird, wird es mit der generischen USB-Umleitung umgeleitet.

Clientordnerumleitung

Feb 17, 2016

Für die Aktivierung und Konfigurierung der Clientordnerumleitung müssen Sie diese für Server und Benutzer aktivieren; auf dem Benutzergerät müssen Sie den umzuleitenden Ordner angeben. Die Benutzeranwendung zur Angabe der Clientordneroptionen ist Teil der Installation von Citrix Receiver, die mit diesem Release bereitgestellt wird.

Wichtig: Die Clientordnerumleitung wird nur auf Windows-Desktopmaschinen unterstützt.

Durch die Clientordnerumleitung ändert sich der Zugriff auf clientseitige Dateien bei der hostseitigen Sitzung. Wird auf dem Server nur die Clientlaufwerkzuordnung aktiviert, werden die clientseitigen vollständigen Volumes den Sitzungen automatisch als UNC-Link (Universal Naming Convention) zugeordnet. Wenn Sie die Clientordnerumleitung auf dem Server aktivieren und der Benutzer sie auf dem Benutzergerät konfiguriert, wird der Teil des vom lokalen Benutzer angegebenen lokalen Volumes umgeleitet.

Nur die vom Benutzer angegebenen Ordner statt des kompletten Dateisystems auf dem Benutzergerät werden als UNC-Links in den Sitzungen angezeigt. Wenn Sie UNC-Links durch die Registrierung deaktivieren, werden Clientordner als zugeordnete Laufwerke in der Sitzung angezeigt.

Die Clientordnerumleitung wird nur auf Windows-Desktopmaschinen unterstützt.

Die Clientordnerumleitung für ein externes USB-Laufwerk wird beim Trennen und Wiederverbinden des Geräts nicht gespeichert.

Systemanforderungen

Serverplattformen, bei denen die Clientordnerumleitung unterstützt wird:

- Windows Server 2008 R2 SP1 (64 Bit)
- Windows Server 2012 (64 Bit)
- Windows Server 2012 R2 (64 Bit)

Clientplattformen, bei denen die Clientordnerumleitung mit neuestem Citrix Receiver unterstützt wird:

- Windows XP (32 Bit und 64 Bit)
- Windows 7 (32 Bit und 64 Bit)
- Windows 8 (32 Bit und 64 Bit)
- Windows 8.1 (32 Bit und 64 Bit)

Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab.

1. Erstellen Sie einen Schlüssel: HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\Client Folder Redirection.
2. Erstellen Sie einen Wert für REG_DWORD.
 - Name: CFROnlyModeAvailable
 - Typ: REG_DWORD
 - Daten: Stellen Sie 1 ein.

Stellen Sie sicher, dass die aktuelle Version von Citrix Receiver installiert ist.

1. Starten Sie CtxCFRUI.exe im Receiver-Installationsverzeichnis.
2. Wählen Sie das Optionsfeld Benutzerdefiniert und fügen Sie Ordner hinzu oder bearbeiten oder entfernen Sie Ordner.
3. Trennen Sie die Sitzungen und stellen Sie dann neue Verbindungen her, damit die Einstellung wirksam wird.

Sicheres Authentifizieren mit Smartcards

Sep 29, 2015

Wenn in Ihrer Organisation Smartcards für die Benutzerauthentifizierung verwendet werden, unterstützt XenApp bzw. XenDesktop die Smartcard-Authentifizierung innerhalb der hier aufgeführten Richtlinien.

Überlegungen zum Verwalten von Smartcards in Ihrer Organisation:

- Mehrere Smartcards und mehrere Leser können an dem gleichen Benutzergerät verwendet werden, wenn jedoch Passthrough-Authentifizierung verwendet wird, kann nur eine Smartcard eingesteckt werden, wenn der Benutzer einen virtuellen Desktop oder eine virtuelle Anwendung startet. Wenn eine Smartcard innerhalb einer Anwendung verwendet wird (z. B. zur digitalen Signierung oder für Verschlüsselungsfunktionen), werden Sie möglicherweise mehrmals zum Einlegen einer Smartcard oder zur Eingabe einer PIN-Nummer aufgefordert. Dieser Fall kann eintreten, wenn eine oder mehrere Smartcards gleichzeitig eingelegt wurden. Wenn Benutzer zum Einlegen einer Smartcard aufgefordert werden und diese sich bereits im Leser befindet, sollten sie auf Abbrechen klicken. Wenn sie zur Eingabe der PIN-Nummer aufgefordert werden, sollten sie diese erneut eingeben.
- Wenn Sie gehostete Anwendungen unter Windows Server 2008 oder 2008 R2 und mit Smartcards verwenden, die den Microsoft-Kryptografiedienstanbieter für Basissmartcards benötigen, werden im Fall der Ausführung einer Smartcard-Transaktion alle anderen Benutzer, die eine Smartcard bei der Anmeldung verwendet haben, blockiert. Weitere Details und einen Hotfix zur Behebung dieses Problems finden Sie unter <http://support.microsoft.com/kb/949538>.

Ihre Organisation hat möglicherweise bestimmte Sicherheitsrichtlinien in Bezug auf die Verwendung von Smartcards. Mit diesen Richtlinien wird z. B. festgelegt, wie Smartcards ausgegeben werden und wie diese von Benutzern gesichert werden sollten. Einige Aspekte dieser Richtlinien müssen ggf. in einer XenApp- bzw. XenDesktop-Umgebung neu bewertet werden.

Sie können PINs mit einem Kartenverwaltungsprogramm oder einem Herstellerdienstprogramm zurücksetzen.

Die folgenden Typen von Smartcardbereitstellungen werden von diesem Release und von gemischten Umgebungen, die diesen Release enthalten, unterstützt. Die Bereitstellungen werden im Detail in anderen Themen in diesem Abschnitt beschrieben. Weitere Konfigurationen funktionieren evtl., werden aber nicht unterstützt.

Typ	Verbindung mit StoreFront
Lokale in Domänen eingebundene Computer	Direkte Verbindung
Remotezugriff von in Domänen eingebundenen Computern	Verbindung über NetScaler Gateway
Nicht in Domänen eingebundene Computer	Direkte Verbindung
Remotezugriff von nicht in Domänen eingebundenen Computern	Verbindung über NetScaler Gateway

Nicht in Domänen eingebundene Computer und Thin Clients mit Zugriff auf die Desktopgerätesite	Verbindung über Desktopgerätesites StoreFront
In Domänen eingebundene Computer und Thin Clients mit Zugriff auf StoreFront über die XenApp Services-URL	Verbindung über XenApp Services-URLs

Die Bereitstellungstypen werden durch die folgenden Merkmale des Benutzergeräts definiert, mit dem der Smartcardleser verbunden ist:

- In Domäne eingebundenes Gerät oder nicht in Domäne eingebundenes Gerät
- Art der Verbindung zwischen Gerät und StoreFront
- Zur Anzeige der virtuellen Desktops und Anwendungen verwendete Software

Darüber hinaus können ebenso smartcardfähige Anwendungen, z. B. Microsoft Word oder Microsoft Excel in diesen Bereitstellungen verwendet werden. Bei diesen Anwendungen können die Benutzer Dokumente digital signieren und verschlüsseln.

Bimodale Authentifizierung

Soweit in der jeweiligen Bereitstellung möglich, unterstützt Receiver die bimodale Authentifizierung, d. h. der Benutzer hat die Wahl, sich mit einer Smartcard oder mit dem Benutzernamen und Kennwort anzumelden. Dies ist nützlich, wenn die Smartcard nicht verwendet werden kann (z. B. sie wurde vom Benutzer zu Hause vergessen oder das Zertifikat ist abgelaufen).

Da Benutzer nicht domänengebundener Geräte sich direkt an Receiver für Windows anmelden, können Sie für diese Benutzer ein Fallback auf die explizite Authentifizierung aktivieren. Wenn Sie die bimodale Authentifizierung konfigurieren, müssen sich Benutzer zuerst mit den Smartcards und PINs anmelden; sie können aber die explizite Authentifizierung auswählen, wenn sie Probleme mit den Smartcards haben.

Wenn Sie NetScaler Gateway bereitstellen, melden sich Benutzer an den Geräten an und werden von Receiver für Windows zur Authentifizierung bei NetScaler Gateway aufgefordert. Dies gilt sowohl für in Domänen eingebundene Geräte als auch für Geräte, die nicht in Domänen eingebunden sind. Benutzer können sich bei NetScaler Gateway mit Smartcard und PIN oder mit expliziten Anmeldeinformationen anmelden. Sie können dann Benutzern die bimodale Authentifizierung für Anmeldungen an NetScaler Gateway bereitstellen. Konfigurieren Sie die Passthrough-Authentifizierung von NetScaler Gateway an StoreFront und delegieren Sie die Validierung der Anmeldeinformationen für Smartcardbenutzer an NetScaler Gateway, sodass Benutzer automatisch bei StoreFront authentifiziert werden.

Überlegungen zu mehreren Active Directory-Gesamtstrukturen

In einer Citrix Umgebung werden Smartcards in einer einzelnen Gesamtstruktur unterstützt. Strukturübergreifende Smartcard-Anmeldungen erfordern eine direkte bidirektionale Gesamtstruktur-Vertrauensstellung für alle Benutzerkonten. Komplexere Mehrfachstruktur-Bereitstellungen mit Smartcards (d. h. Vertrauensstellungen sind nur unidirektional oder sonstiger Art) werden nicht unterstützt.

Sie können Smartcards in einer Citrix Umgebung mit Remotedesktops verwenden. Dieses Feature kann lokal installiert werden (auf dem Benutzergerät, mit dem die Smartcard verbunden ist) oder remote (auf dem Remotedesktop, mit dem das Benutzergerät verbunden wird).

Richtlinie zum Entfernen der Smartcard

Die Richtlinie zum Entfernen der Smartcard legt fest, was passiert, wenn die Smartcard während einer Sitzung entfernt wird. Die Richtlinie zum Entfernen der Smartcard wird im Windows-Betriebssystem konfiguriert und verarbeitet.

Richtlinieneinstellung	Desktop-Verhalten
Keine Aktion	Keine Aktion.
Arbeitsstation sperren	Die Desktopsitzung wird getrennt und der virtuelle Desktop gesperrt.
Abmeldung erzwingen	Der Benutzer wird zur Abmeldung gezwungen. Wenn die Netzwerkverbindung unterbrochen ist und diese Einstellung aktiviert wird, wird die Sitzung möglicherweise abgemeldet und der Benutzer verliert Daten.
Trennen bei einer Remotedienstesitzung	Die Sitzung wird getrennt und der virtuelle Desktop gesperrt.

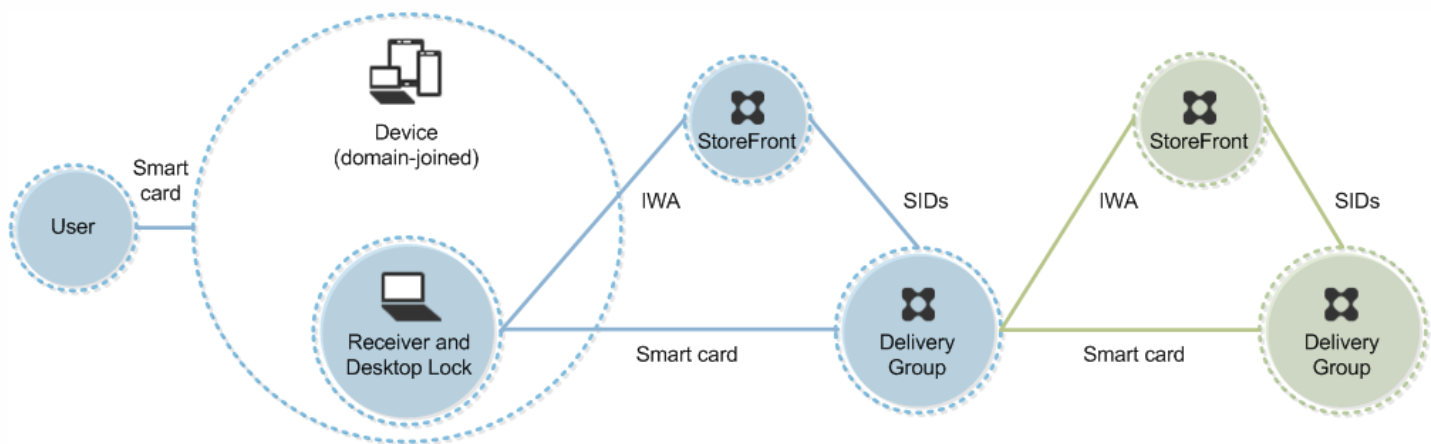
Überprüfen der Zertifikatsperrlisten

Wenn die Überprüfung von Zertifikatsperrlisten aktiviert ist und ein Benutzer führt eine Smartcard mit einem ungültigen Zertifikat in einen Smartcardleser ein, kann der Benutzer nicht authentifiziert werden oder nicht auf den mit dem Zertifikat verbundenen Desktop oder die Anwendung zugreifen. Bei einem ungültigen Zertifikat für die E-Mail-Entschlüsselung bleibt die E-Mail beispielsweise verschlüsselt. Wenn andere Zertifikate auf der Smartcard, z. B. solche, die für die Authentifizierung verwendet werden, noch gültig sind, bleiben diese Funktionen weiterhin aktiv.

Beispiele

Im Folgenden sind einige Beispiele der Smartcardbereitstellung aufgeführt.

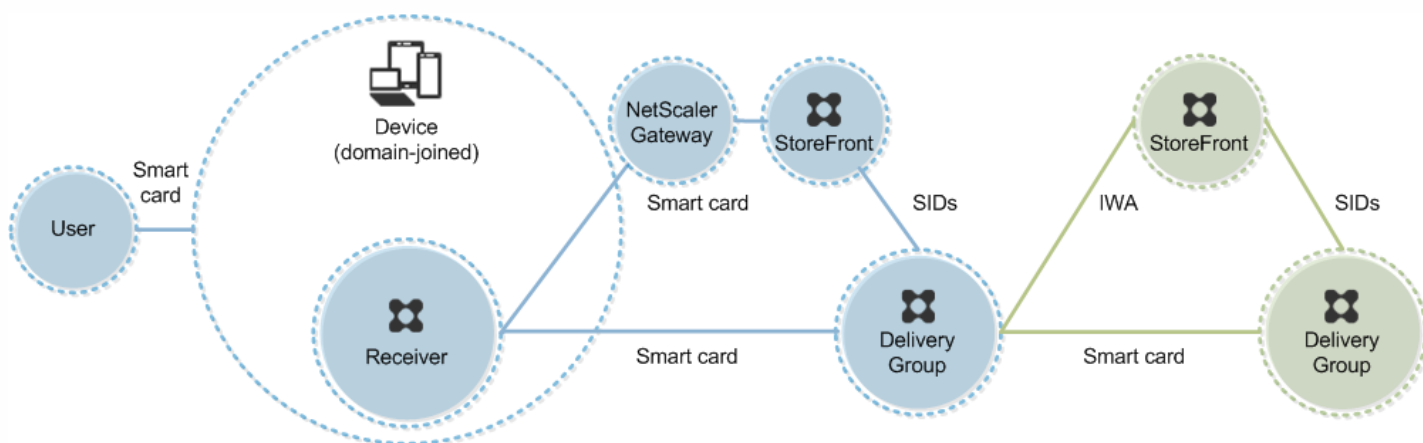
Diese Bereitstellung bezieht sich auf in Domänen eingebundene Benutzergeräte mit Desktop Viewer und Direktverbindung mit StoreFront.



Zum Anmelden beim Gerät benötigt der Benutzer eine Smartcard und eine PIN. Der Benutzer wird dann durch Receiver beim StoreFront-Server mittels integrierter Windows-Authentifizierung (IWA) authentifiziert. StoreFront übergibt die Sicherheits-IDs (SIDs) an XenApp bzw. XenDesktop. Wenn der Benutzer einen virtuellen Desktop oder eine Anwendung startet, wird er nicht aufgefordert, die PIN neu einzugeben, da in Receiver das Feature "Single Sign-On" konfiguriert ist.

Diese Bereitstellung kann um einen zweiten StoreFront-Server und einen Server mit gehosteten Anwendungen auf ein Double-Hop-System erweitert werden. Ein Receiver auf dem virtuellen Desktop übernimmt die Authentifizierung beim zweiten StoreFront-Server. Für diese zweite Verbindung kann eine beliebige Authentifizierungsmethode verwendet werden. Die für den ersten Hop dargestellte Konfiguration kann im zweiten Hop wiederverwendet oder nur für den zweiten Hop verwendet werden.

Diese Bereitstellung bezieht sich auf in Domänen eingebundene Benutzergeräte mit Desktop Viewer und Verbindung mit StoreFront über NetScaler Gateway/Access Gateway.

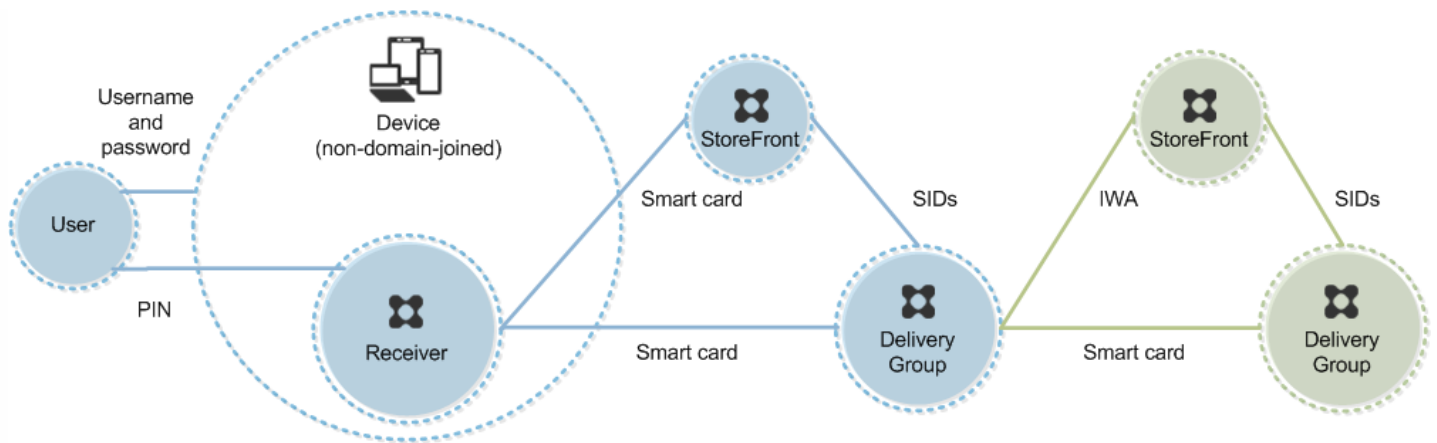


Der Benutzer meldet sich mit Smartcard und PIN beim Gerät und anschließend erneut bei NetScaler Gateway oder Access Gateway an. Die zweite Anmeldung kann entweder mit Smartcard und PIN oder einem Benutzernamen und einem Kennwort erfolgen, da Receiver in dieser Bereitstellung eine bimodale Authentifizierung zulässt.

Der Benutzer wird automatisch bei StoreFront angemeldet; StoreFront übergibt die Sicherheits-IDs (SIDs) an XenApp bzw. XenDesktop. Wenn der Benutzer einen virtuellen Desktop oder eine Anwendung startet, wird er nicht aufgefordert, die PIN neu einzugeben, da in Receiver das Feature "Single Sign-On" konfiguriert ist.

Diese Bereitstellung kann um einen zweiten StoreFront-Server und einen Server mit gehosteten Anwendungen auf ein Double-Hop-System erweitert werden. Ein Receiver auf dem virtuellen Desktop übernimmt die Authentifizierung beim zweiten StoreFront-Server. Für diese zweite Verbindung kann eine beliebige Authentifizierungsmethode verwendet werden. Die für den ersten Hop dargestellte Konfiguration kann im zweiten Hop wiederverwendet oder nur für den zweiten Hop verwendet werden.

Diese Bereitstellung bezieht sich auf nicht in Domänen eingebundene Benutzergeräte mit Desktop Viewer und Direktverbindung mit StoreFront.

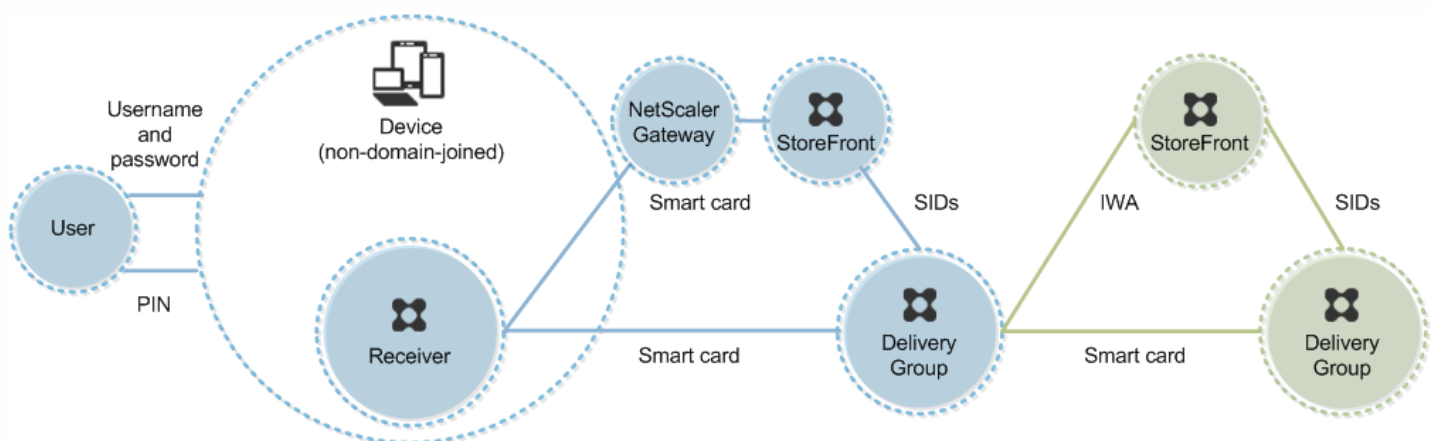


Ein Benutzer meldet sich beim Gerät an. Normalerweise muss er seinen Benutzernamen und das Kennwort eingeben, aber da das Gerät nicht Mitglied einer Domäne ist, sind die Anmeldeinformationen für diese Anmeldung optional. Da bimodale Authentifizierung in dieser Bereitstellung möglich ist, fordert Receiver den Benutzer auf, sich entweder mit Smartcard und PIN oder mit Benutzernamen und Kennwort anzumelden. Receiver authentifiziert dann bei StoreFront.

StoreFront übergibt die Sicherheits-IDs (SIDs) an XenApp bzw. XenDesktop. Wenn der Benutzer einen virtuellen Desktop oder eine Anwendung startet, wird er aufgefordert, die PIN neu einzugeben, da Single Sign-On in dieser Bereitstellung nicht verfügbar ist.

Diese Bereitstellung kann um einen zweiten StoreFront-Server und einen Server mit gehosteten Anwendungen auf ein Double-Hop-System erweitert werden. Ein Receiver auf dem virtuellen Desktop übernimmt die Authentifizierung beim zweiten StoreFront-Server. Für diese zweite Verbindung kann eine beliebige Authentifizierungsmethode verwendet werden. Die für den ersten Hop dargestellte Konfiguration kann im zweiten Hop wiederverwendet oder nur für den zweiten Hop verwendet werden.

Diese Bereitstellung bezieht sich auf nicht in Domänen eingebundene Benutzergeräte mit Desktop Viewer und Direktverbindung mit StoreFront.



Ein Benutzer meldet sich beim Gerät an. Normalerweise muss er seinen Benutzernamen und das Kennwort eingeben, aber da das Gerät nicht Mitglied einer Domäne ist, sind die Anmeldeinformationen für diese Anmeldung optional. Da bimodale Authentifizierung in dieser Bereitstellung möglich ist, fordert Receiver den Benutzer auf, sich entweder mit Smartcard und PIN oder mit Benutzernamen und Kennwort anzumelden. Receiver authentifiziert dann bei StoreFront.

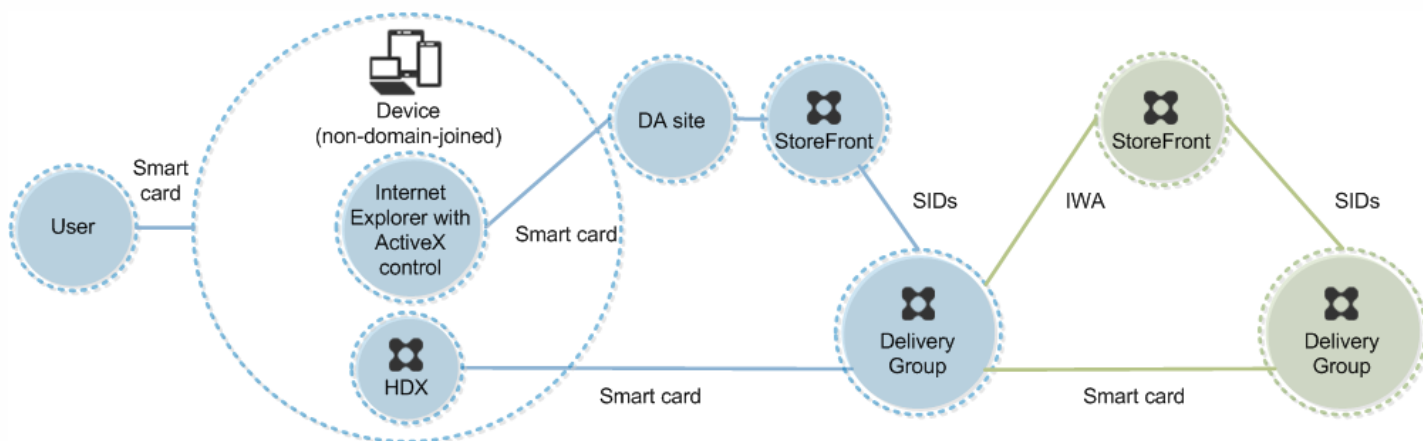
StoreFront übergibt die Sicherheits-IDs (SIDs) an XenApp bzw. XenDesktop. Wenn der Benutzer einen virtuellen Desktop oder eine Anwendung startet, wird er aufgefordert, die PIN neu einzugeben, da Single Sign-On in dieser Bereitstellung nicht verfügbar ist.

Diese Bereitstellung kann um einen zweiten StoreFront-Server und einen Server mit gehosteten Anwendungen auf ein Double-Hop-System erweitert werden. Ein Receiver auf dem virtuellen Desktop übernimmt die Authentifizierung beim zweiten StoreFront-Server. Für diese zweite Verbindung kann eine beliebige Authentifizierungsmethode verwendet werden. Die für den ersten Hop dargestellte Konfiguration kann im zweiten Hop wiederverwendet oder nur für den zweiten Hop verwendet werden.

Diese Bereitstellung bezieht sich auf nicht in Domänen eingebundene Benutzergeräte, auf denen möglicherweise Desktop Lock ausgeführt wird und die mit StoreFront über Desktopgerätesites verbunden werden.

Desktop Lock ist eine eigenständige Komponente, die mit XenApp, XenDesktop und VDI-in-a-Box auf den Markt gebracht wurde. Das Programm ist eine Alternative zu Desktop Viewer und wird hauptsächlich für umfunktionierte Windows-Computer und Thin Clients verwendet. Desktop Lock ersetzt die Windows-Shell und Task-Manager bei diesen Benutzergeräten, wodurch der Benutzerzugriff auf die zugrunde liegenden Geräte verhindert wird. Mit Desktop Lock können Benutzer auf Desktops von Windows-Servermaschinen und Windows-Desktopmaschinen zugreifen.

Hinweis: Die Installation von Desktop Lock ist optional.



Zum Anmelden beim Gerät benötigt der Benutzer eine Smartcard. Wenn Desktop Lock auf dem Gerät ausgeführt wird, wird das Gerät so konfiguriert, dass eine Desktopgerätesite über Internet Explorer im Kioskmodus gestartet wird. Der Benutzer wird durch ein ActiveX-Steuerelement der Site aufgefordert, seine PIN einzugeben, die dann an StoreFront gesendet wird. StoreFront übergibt die Sicherheits-IDs (SIDs) an XenApp bzw. XenDesktop. Der erste verfügbare Desktop in der alphabetischen Liste einer zugewiesenen Desktopgruppe wird gestartet.

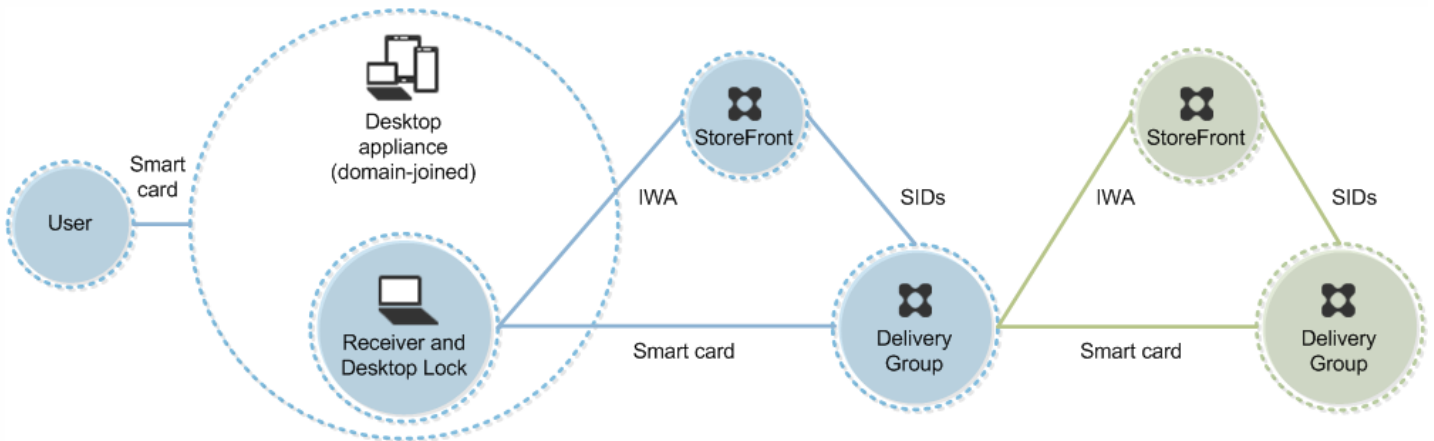
Diese Bereitstellung kann um einen zweiten StoreFront-Server und einen Server mit gehosteten Anwendungen auf ein Double-Hop-System erweitert werden. Ein Receiver auf dem virtuellen Desktop übernimmt die Authentifizierung beim zweiten StoreFront-Server. Für diese zweite Verbindung kann eine beliebige Authentifizierungsmethode verwendet werden. Die für den ersten Hop dargestellte Konfiguration kann im zweiten Hop wiederverwendet oder nur für den zweiten Hop verwendet werden.

Diese Bereitstellung bezieht sich auf in Domänen eingebundene Benutzergeräte, auf denen Desktop Lock ausgeführt wird

und die mit StoreFront über XenApp Services-URLs verbunden werden.

Desktop Lock ist eine eigenständige Komponente, die mit XenApp, XenDesktop und VDI-in-a-Box auf den Markt gebracht wurde. Das Programm ist eine Alternative zu Desktop Viewer und wird hauptsächlich für umfunktionierte Windows-Computer und Thin Clients verwendet. Desktop Lock ersetzt die Windows-Shell und Task-Manager bei diesen Benutzergeräten, wodurch der Benutzerzugriff auf die zugrunde liegenden Geräte verhindert wird. Mit Desktop Lock können Benutzer auf Desktops von Windows-Servermaschinen und Windows-Desktopmaschinen zugreifen.

Hinweis: Die Installation von Desktop Lock ist optional.



Zum Anmelden beim Gerät benötigt der Benutzer eine Smartcard und eine PIN. Wenn Desktop Lock auf dem Gerät ausgeführt wird, wird der Benutzer beim Storefront-Server über die integrierte Windows-Authentifizierung (IWA) authentifiziert. StoreFront übergibt die Sicherheits-IDs (SIDs) an XenApp bzw. XenDesktop. Wenn der Benutzer einen virtuellen Desktop startet, wird er nicht aufgefordert, die PIN neu einzugeben, da in Receiver Single Sign-On konfiguriert ist.

Diese Bereitstellung kann um einen zweiten StoreFront-Server und einen Server mit gehosteten Anwendungen auf ein Double-Hop-System erweitert werden. Ein Receiver auf dem virtuellen Desktop übernimmt die Authentifizierung beim zweiten StoreFront-Server. Für diese zweite Verbindung kann eine beliebige Authentifizierungsmethode verwendet werden. Die für den ersten Hop dargestellte Konfiguration kann im zweiten Hop wiederverwendet oder nur für den zweiten Hop verwendet werden.

Passthrough-Authentifizierung

Die Passthrough-Authentifizierung mit Smartcards bei virtuellen Desktops wird auf Benutzergeräten unterstützt, auf denen die folgenden Betriebssysteme ausgeführt werden:

- Windows 8
- Windows 7 SP1, Enterprise und Professional Edition
- Windows XP Service Pack 3, 32-Bit-Edition

Die Passthrough-Authentifizierung mit Smartcards bei gehosteten Anwendungen wird auf Servern unterstützt, auf denen die folgenden Betriebssysteme ausgeführt werden:

- Windows Server 2012
- Windows Server 2008

Wenn Sie die Passthrough-Authentifizierung mit Smartcards für gehostete Anwendungen verwenden, stellen Sie sicher, dass Sie für Passthrough mit Smartcard als Authentifizierungsmethode für die Site die Verwendung von Kerberos aktivieren.

Hinweis: Die Verfügbarkeit der Passthrough-Authentifizierung mit Smartcards hängt von vielen Faktoren ab, u. a.:

- Sicherheitsrichtlinien für die Passthrough-Authentifizierung der jeweiligen Organisation
- Typ und Konfiguration der Middleware
- Typen der Smartcardleser
- Richtlinie für das Zwischenspeichern von Middleware-PINs

Die Passthrough-Authentifizierung mit Smartcards wird in Citrix StoreFront konfiguriert. Informationen finden Sie unter [Konfigurieren des Authentifizierungsdiensts](#).

Single Sign-On

Single Sign-On ist ein Citrix Feature, mit dem die Passthrough-Authentifizierung in Starts von virtuellen Desktops und Anwendungen implementiert wird. Sie können dieses Feature bei Smartcardbereitstellungen verwenden, die in Domänen eingebunden und direkt mit StoreFront verbunden sind, sowie bei in Domänen eingebundenen und über NetScaler mit StoreFront verbundenen Bereitstellungen. So müssen Benutzer ihre PIN weniger häufig eingeben. Zur Verwendung von Single Sign-On in diesen Bereitstellungstypen bearbeiten Sie die folgenden Parameter in default.ica. Diese Datei ist auf dem StoreFront-Server:

- **In Domänen eingebundene, direkt mit StoreFront verbundene Smartcard-Bereitstellungen:** Einstellung für DisableCtrlAltDel auf Off
- **In Domänen eingebundene, über NetScaler mit StoreFront verbundene Smartcard-Bereitstellungen:** Einstellung für UseLocalUserAndPassword auf On

Weitere Anweisungen zum Einrichten dieser Parameter finden Sie in der Dokumentation für StoreFront oder NetScaler Gateway.

Die Verfügbarkeit der Single Sign-On-Funktion hängt von vielen Faktoren ab, u. a.:

- Sicherheitsrichtlinien für Single Sign-On der jeweiligen Organisation
- Typ und Konfiguration der Middleware
- Typen der Smartcardleser
- Richtlinie für das Zwischenspeichern von Middleware-PINs

Hinweis: Wenn Benutzer sich beim Virtual Delivery Agent (VDA) anmelden und es ist ein Smartcardleser angeschlossen, wird möglicherweise eine Windows-Kachel angezeigt, die die letzte erfolgreiche Authentifizierungsmethode repräsentiert, z. B. Smartcard oder Kennwort. Daher wird bei aktiviertem Single Sign-On ggf. eine entsprechende Kachel angezeigt. Zur Anmeldung muss der Benutzer über Benutzer wechseln statt der Single Sign-On-Kachel eine andere Kachel auswählen, da die Single Sign-On-Kachel nicht funktioniert.

Systemanforderungen für Smartcards

Sep 29, 2015

Smartcardleser vom Typ ZKA (Zentraler Kreditausschuss) Klasse 1, die USB Chip-/CCID-konform (Smart Card Interface Devices) sind, werden unterstützt. Diese enthalten einen Schlitz, in den die Smartcard eingeführt wird. Andere Klassen, einschließlich Klasse 2 (Leser mit Tastatur für die Eingabe von PINs), kontaktlose Leser und virtuelle Smartcards, die auf dem TPM-Chip (Trusted Platform Module) basieren, werden nicht unterstützt.

Für den Smartcardleser muss ein Gerätetreiber auf dem Benutzergerät installiert werden. Viele Smartcardleser können mit dem von Microsoft bereitgestellten CCID-Gerätetreiber benutzt werden.

Sie müssen einen Gerätetreiber und Kryptografiedienstbietersoftware (CSP) vom Smartcard-Hersteller beziehen und beides auf Benutzergeräten und auf virtuellen Desktops installieren. Der Treiber und die CSP-Software müssen mit XenApp bzw. XenDesktop kompatibel sein. Überprüfen Sie die Kompatibilität anhand der Hersteller-Dokumentation. Citrix empfiehlt folgende Vorgehensweise:

- Installieren von Treiber und CSP-Software vor der Installation von Citrix Software
- Installieren und Testen der Treiber auf einem physischen Computer vor der Installation von Citrix Software

Hinweis: Für virtuelle Desktops unter Windows 7 mit Smartcards, die das Minitreibermodell unterstützen und verwenden, werden die Smartcard-Minitreiber automatisch heruntergeladen. Die Treiber können auch über <http://catalog.update.microsoft.com> oder den Hersteller bezogen werden. Wird PKCS#11-Middleware benötigt, wenden Sie sich an den Smartcardhersteller.

Die Unterstützung von Smartcards umfasst außerdem Komponenten, die über Citrix Partner bezogen werden können. Diese werden von den jeweiligen Partnern aktualisiert und in den vorliegenden Abschnitten nicht erläutert. Weitere Informationen finden Sie unter <http://www.citrix.com/ready/>.

- Smartcards werden nur für den Remotezugriff auf physische Büro-PCs mit Windows 7 oder Windows 8 unterstützt; Smartcards werden nicht für Büro-PCs mit Windows XP unterstützt.
- Die folgenden Smartcards wurden mit Remote-PC-Zugriff getestet:
 - Gemalto .Net 2.0 mit Gemalto .Net-Minitreiber
 - Gemalto IDPrime .NET 510 mit Gemalto .Net-Minitreiber
 - Gemalto PIV-Karten mit ActivIdentity ActivClient 6.2

Auf Benutzergeräten muss Citrix Receiver, geeignete Middleware und eines der folgenden Betriebssysteme ausgeführt werden:

- Windows 8 (einschließlich Embedded Edition), 32-Bit- und 64-Bit-Editionen
- Windows 7 (einschließlich Embedded Edition), 32-Bit- und 64-Bit-Editionen
- Windows XP Professional, Service Pack 3 (32-Bit-Edition)

Die Smartcard-Unterstützung in Receiver basiert auf dem PC/SC-Standard (Personal Computer/Smart Card) von Microsoft.

Als Mindestanforderung müssen Smartcards und Smartcardleser vom zu Grunde liegenden Windows-Betriebssystem unterstützt werden und vom Microsoft Windows Hardware Quality Labs (WHQL) für die Verwendung auf Computern mit einem qualifizierenden Windows-Betriebssystem zugelassen sein. Weitere Informationen zur PC/SC-Kompatibilität der Hardware finden Sie unter <http://www.microsoft.com>.

Die folgenden Kombinationen aus Smartcard und Middleware wurden von Citrix als repräsentatives Beispiel ihres Typs getestet. Es können jedoch auch andere Smartcards und Middleware verwendet werden. Weitere Informationen über Citrix-kompatible Smartcards und Middleware finden Sie unter <http://www.citrix.com/ready>.

Windows		
Middleware	Geeignete Karten	Hinweise
ActivClient 6.2 (DoD CAC Edition) im Modus GSC-IS	DoD CAC-Karte	
ActivClient 7.0 im Modus GSC-IS	DoD CAC-Karte	
ActivClient 7.0 im Modus PIV	DoD CAC-Karte, NIST PIV-Karte	
GemAlto-Minitreiber für .NET-Karte	GemAlto IDPrime .NET 510	
SafeNet Authentication Client 8.x für Windows	USB eToken 5100	

Führen Sie die folgenden zusätzlichen Aufgaben vor der Smartcard-Bereitstellung aus:

- Stellen Sie sicher, dass die Public Key-Infrastruktur entsprechend konfiguriert ist. Hierzu gehört, dass die Zertifikat-zu-Konto-Zuordnung richtig für die Active Directory-Umgebung konfiguriert ist, und dass die Validierung des Benutzerzertifikats ausgeführt werden kann.
- Konfigurieren Sie die Komponenten, die TLS 1.0 verwenden, für die Smartcard-Anmeldung.
- Stellen Sie sicher, dass die Bereitstellung die Systemanforderungen der anderen Citrix Komponenten erfüllt, die mit Smartcards verwendet werden, u. a. Receiver und StoreFront.

Aktivieren der Smartcard-Verwendung

Sep 29, 2015

Zum Aktivieren der Smartcard-Verwendung benötigen Sie Zugriff auf die Ihren Sites zugeordneten Server, auf denen Folgendes ausgeführt wird:

- Active Directory-Domänencontroller für das Benutzerkonto mit zugeordnetem Anmeldezertifikat auf der Smartcard
- Delivery Controller
- Citrix StoreFront
- Citrix NetScaler Gateway/Citrix Access Gateway 10.x
- Virtual Delivery Agent
- Optional für Remotezugriff: Microsoft Exchange Server

Hinweis: Für Remote-PC-Zugriff werden Smartcards nur auf physischen Computern unter Windows 7 oder Windows 8 unterstützt. Smartcards werden für physische Computer unter Windows XP oder Vista nicht unterstützt.

1. Machen Sie sich mit der Smartcard-Technik im allgemeinen und der Ihrer eigenen Smartcard vertraut.
2. Informieren Sie sich über die Installation und Pflege von Zertifikaten in verteilten Umgebungen.
3. Machen Sie sich mit der Funktionsweise und der Dokumentation folgender Anwendungen vertraut:
 1. Citrix StoreFront 2.0
 2. Citrix NetScaler Gateway/Citrix Access Gateway 10.x
Hinweis: Diese werden für die meisten Smartcard-Bereitstellungen benötigt.
 3. Citrix Receiver für Windows 4.0 und 3.4
 4. Citrix SDK
4. Aktivieren Sie das Produkt für die Verwendung von Smartcards.
 1. Geben Sie die Smartcards an die Benutzer aus und berücksichtigen Sie dabei die Kartenausstellungsrichtlinie.
 2. (Optional): Richten Sie Smartcards ein, damit Benutzer Remote-PC-Zugriff verwenden können.
 3. Installieren und konfigurieren Sie den Delivery Controller und StoreFront (falls nicht bereits für Smartcardremoting installiert).
5. Aktivieren Sie StoreFront für die Verwendung mit Smartcard. Weitere Informationen finden Sie unter [Konfigurieren der Smartcard-Authentifizierung](#).
6. Aktivieren Sie NetScaler Gateway/Access Gateway für die Verwendung mit Smartcard. Weitere Informationen finden Sie unter [Configuring Authentication and Authorization](#) und [Configuring Smart Card Access with the Web Interface](#).
7. Aktivieren Sie den Virtual Delivery Agent (VDA) für die Verwendung mit Smartcard. Weitere Informationen über das Installieren und Konfigurieren des VDAs finden Sie unter [Installieren über die grafische Oberfläche](#).
 1. Stellen Sie sicher, dass der Virtual Delivery Agent die erforderlichen Anwendungen und Updates hat.
 2. Installieren Sie die Middleware.
 3. Richten Sie Smartcardremoting ein, damit die Kommunikation von Smartcarddaten zwischen Receiver auf einem Benutzergerät und einer virtuellen Desktopsitzung möglich ist.
8. Aktivieren Sie die Benutzergeräte für die Verwendung von Smartcards. Weitere Informationen finden Sie unter [Konfigurieren der Smartcard-Authentifizierung](#).

Benutzergeräte unter Windows

Zu diesen Benutzergeräten zählen in Domänen eingebundene oder nicht in Domänen eingebundene Maschinen.

1. Importieren Sie das Zertifizierungsstellen-Stammzertifikat und das Zertifikat der ausstellenden Zertifizierungsstelle in den Schlüsselspeicher des Geräts.
2. Installieren Sie die kryptografische Middleware.

3. Installieren und konfigurieren Sie Receiver für Windows. Stellen Sie sicher, dass Sie `icaclient.adm` mit der Gruppenrichtlinien-Verwaltungskontrolle importieren und die Smartcard-Authentifizierung aktivieren.
 - Installieren Sie Receiver für Windows Enterprise 3.4 auf Thin Clients und Computern mit Desktop Lock.
 - Installieren Sie Receiver für Windows 4.0 auf allen anderen Geräten.
9. Testen Sie Ihre Bereitstellung.

Stellen Sie sicher, dass die Bereitstellung richtig konfiguriert ist, indem Sie den virtuellen Desktop mit der Smartcard eines Testbenutzers starten. Testen Sie alle möglichen Zugriffsmechanismen (beispielsweise Zugriff auf den Desktop über Internet Explorer und Receiver).

Überwachen und Problembehandlung

Sep 29, 2015

Director ist ein webbasiertes Tool, mit dem die Support- und Helpdesk-Teams eine XenApp- oder XenDesktop-Umgebung überwachen, potenziell systembedrohende Probleme rechtzeitig behandeln und Support für Endbenutzer leisten können.

Director bietet verschiedene Ansichten der Schnittstelle, die auf bestimmte Administratoren abgestimmt sind. Produktberechtigungen bestimmen, was angezeigt wird und welche Befehle verfügbar sind.

Beispiel: Helpdeskadministratoren sehen eine auf Helpdeskaufgaben abgestimmte Schnittstelle. Director ermöglicht Helpdeskadministratoren, nach dem Benutzer zu suchen, der das Problem gemeldet hat, und die diesem Benutzer zugeordneten Aktivitäten anzuzeigen, z. B. den Status der Anwendungen und Prozesse des Benutzers. So können Probleme schnell gelöst werden, indem Aktionen wie z. B. das Beenden einer nicht reagierenden Anwendung oder eines Prozesses, das Spiegeln von Vorgängen auf der Maschine des Benutzers, der Neustart der Maschine oder das Zurücksetzen des Benutzerprofils durchgeführt werden.

Im Gegensatz dazu sehen und verwalten Volladministratoren die gesamte Website und können Befehle für mehrere Benutzer und Maschinen ausführen. Das Dashboard bietet einen Überblick über die wichtigsten Aspekte einer Bereitstellung, z. B. den Status von Sitzungen und Benutzeranmeldungen und die Infrastruktur der Site. Die Informationen werden jede Minute aktualisiert. Wenn Probleme auftreten, werden automatisch Details zu Anzahl und Art der Fehler angezeigt.

CtxPvdDiag.exe ist ein Diagnosetool, mit dem Sie die an persönlichen vDisks vorgenommenen Änderungen überwachen können.

Zusätzlich zur Anzeige von Verlaufsdaten in der Director-Konsole können Sie Daten mit der Überwachungsdienst-OData-API abfragen. Sie können die API zum Analysieren historischer Trends für die Planung oder für die gezielte Problembehandlung bei Verbindungs- und Maschinenfehlern einsetzen. Sie können die API auch zum Extrahieren von Informationen für die Eingabe in andere Tools und Prozesse verwenden.

Überwachen von Umgebungen mit Director

Sep 29, 2015

Director ist standardmäßig als Website auf dem Delivery Controller installiert. Informationen zu Voraussetzungen und anderen Details finden Sie in den Systemanforderungen für dieses Release.

Dieses Release von Director ist nicht kompatibel mit XenApp-Bereitstellungen vor Version 7.5 und XenDesktop-Bereitstellungen vor Version 7.

Wenn Director in einer Umgebung mit mehreren Sites verwendet wird, synchronisieren Sie die Systemuhren auf allen Servern, auf denen Controller, Director und andere wichtige Kernkomponenten installiert sind. Ansonsten werden die Sites in Director möglicherweise nicht richtig angezeigt.

Tipp: Wenn Sie beabsichtigen, neben XenApp 7.5- und XenDesktop 7.x-Sites auch XenApp 6.5-Sites zu überwachen, empfiehlt Citrix, Director und die Director-Konsole, die zur Überwachung von XenApp 6.5-Sites verwendet wird, auf separaten Servern zu installieren.

Wichtig: Zum Schutz von als Nur-Text über das Netzwerk gesendeten Benutzernamen und Kennwörtern empfiehlt Citrix dringend, nur Director-Verbindungen mit HTTPS und nicht mit HTTP zuzulassen. Bestimmte Tools können Nur-Text-Benutzernamen und -Kennwörter in (unverschlüsselten) HTTP-Netzwerkpaketen lesen, wodurch ein Sicherheitsrisiko für Benutzer entsteht.

Konfigurieren von Berechtigungen

Um eine Anmeldung bei Director vornehmen zu können, müssen Administratoren mit den Berechtigungen für Director Active Directory-Domänenbenutzer sein und die folgenden Berechtigungen haben:

- Leseberechtigungen in allen zu durchsuchenden Active Directory-Gesamtstrukturen (siehe [Erweiterte Konfiguration](#))
- Konfigurierte delegierte Administratorrollen (siehe [Delegierte Administration und Director](#))
- Zum Spiegeln von Benutzern muss für Administratoren eine Microsoft-Gruppenrichtlinie für Windows-Remoteunterstützung konfiguriert werden. Darüber hinaus gilt Folgendes:
 - Bei der Installation von VDAs stellen Sie sicher, dass die Windows-Remoteunterstützung auf allen Benutzergeräten aktiviert ist (standardmäßig aktiviert).
 - Wenn Sie Director auf einem Server installieren, stellen Sie sicher, dass die Windows-Remoteunterstützung installiert ist (standardmäßig ausgewählt). Allerdings ist sie auf dem Server standardmäßig deaktiviert. Das Feature muss für Director nicht aktiviert werden, um Benutzern zu helfen. Citrix empfiehlt, das Feature deaktiviert zu lassen, um die Sicherheit auf dem Server zu erhöhen.
 - Damit Administratoren die Windows-Remoteunterstützung initiieren können, müssen Sie ihnen mit den entsprechenden Einstellungen der Microsoft-Gruppenrichtlinie die Berechtigungen für die Remoteunterstützung erteilen. Informationen finden Sie unter [CTX127388: How to Enable Remote Assistance for Desktop Director](#).
- Bei Benutzergeräten mit VDA-Versionen vor 7 ist eine zusätzliche Konfiguration erforderlich. Informationen hierzu finden Sie unter [Konfigurieren von Berechtigungen für VDAs vor XenDesktop 7](#).

Installieren von Director

Installieren Sie Director mit dem Installer. Dieser prüft, ob die Voraussetzungen erfüllt sind, installiert fehlende Komponenten, richtet die Director-Website ein und führt die Grundkonfiguration durch. Die Standardkonfiguration, die der Installer bietet, eignet sich für typische Bereitstellungen. Fügen Sie Director mit dem Installationsprogramm hinzu, falls dies während der Installation nicht geschehen ist. Zum Hinzufügen zusätzlicher Komponenten führen Sie den Installer erneut aus und wählen Sie die zu installierenden Komponenten aus. Informationen zur Verwendung des Installationsprogramms

finden Sie in den Abschnitten zur Installation. Citrix empfiehlt, dass Sie die Installation ausschließlich mit dem Installer des Produkts und nicht über die MSI-Datei durchführen.

Wenn Director auf dem Controller installiert ist, erfolgt automatisch eine Konfiguration mit localhost als Serveradresse und Director kommuniziert standardmäßig mit dem lokalen Controller.

Zur Installation von Director auf einem dedizierten, Controller-remoten Server werden Sie zur Eingabe des FQDN oder der IP-Adresse eines Controllers aufgefordert. Director kommuniziert standardmäßig mit diesem angegebenen Controller. Geben Sie nur eine Controlleradresse für jede zu überwachende Site ein. Director ermittelt automatisch alle anderen Controller in derselben Site und wechselt zu diesen anderen Controllern, wenn der von Ihnen angegebene Controller ausfällt.

Hinweis: Director führt keinen Lastausgleich zwischen Controllern aus.

Citrix empfiehlt die Implementierung von SSL auf der IIS-Website, die Director hostet, um die Kommunikation zwischen dem Browser und dem Webserver zu schützen. In der Dokumentation von Microsoft zu IIS finden Sie entsprechende Anweisungen. Zum Aktivieren von SSL ist keine Director-Konfiguration erforderlich.

Anmelden an Director

Die Director-Website ist unter https oder http:///Director.

Wenn eine der Sites einer Bereitstellung mit mehreren Sites ausfällt, dauert die Anmeldung für Director etwas länger, während Verbindungsversuche mit dieser Site laufen.

Bei der delegierten Administration werden drei Konzepte eingesetzt: Administratoren, Rollen und Geltungsbereiche. Berechtigungen richten sich nach der Administratorrolle und dem Geltungsbereich dieser Rolle. Beispiel: Einem Administrator wird die Helpdeskadministratorrolle zugewiesen, bei der der Geltungsbereich die Verantwortung für Endbenutzer an nur einer Site umfasst.

Weitere Informationen über das Erstellen von delegierten Administratoren finden Sie im Hauptabschnitt [Delegierte Administration](#).

Durch die administrativen Berechtigungen wird festgelegt, wie die Director-Benutzeroberfläche für Administratoren dargestellt wird und welche Aufgaben sie ausführen können. Mit Berechtigungen wird Folgendes festgelegt:

- Die Seiten, auf die der Administrator zugreifen kann, kollektiv als "Ansicht" bezeichnet
- Die Desktops, Maschinen und Sitzungen, die der Administrator anzeigen und verwenden kann
- Die Befehle, die der Administrator ausführen kann, z. B. das Spiegeln einer Benutzersitzung oder das Aktivieren des Wartungsmodus

Über die integrierten Rollen und Berechtigungen wird außerdem gesteuert, wie Administratoren Director verwenden:

Administratorrolle	Berechtigungen in Director
Volladministrator	Hat vollständigen Zugriff auf alle Ansichten und kann alle Befehle ausführen, einschließlich Spiegeln einer Benutzersitzung, Aktivieren des Wartungsmodus und Exportieren von Trenddaten.
Bereitstellungsgruppenadministrator	Hat vollständigen Zugriff auf alle Ansichten und kann alle Befehle ausführen, einschließlich Spiegeln einer Benutzersitzung, Aktivieren des Wartungsmodus und

Administratorrolle	Exportieren von Trenddaten. Berechtigungen in Director
Lesezugriffadministrator	Kann auf alle Ansichten zugreifen und alle Objekte in angegebenen Geltungsbereichen sowie globale Informationen anzeigen. Kann Berichte aus HDX-Kanälen herunterladen und Trenddaten mit der Exportoption in der Ansicht "Trends" exportieren. Kann keine anderen Befehle ausführen oder Daten in den Ansichten ändern.
Helpdeskadministrator	Kann nur auf die Ansichten "Helpdesk" und "Benutzerdetails" zugreifen und nur Objekte anzeigen, die dem Administrator zur Verwaltung übertragen wurden. Kann eine Benutzersitzung spiegeln und Befehle für diesen Benutzer ausführen. Kann Vorgänge im Wartungsmodus ausführen. Kann Energieoptionen auf Desktopbetriebssystemmaschinen verwenden. Kann nicht auf das Dashboard, Trends oder Filteransichten zugreifen. Kann keine Energieoptionen auf Serverbetriebssystemmaschinen verwenden.
Maschinenkatalogadministrator	Kein Zugriff. Dieser Administrator wird für Director nicht unterstützt und er kann keine Daten anzeigen.
Hostadministrator	Kein Zugriff. Dieser Administrator wird für Director nicht unterstützt und er kann keine Daten anzeigen.

Hinweis: Sie können über delegierte Administratorrollen und -berechtigungen die Ansichten in Director ändern, z. B. Schaltflächen entfernen oder aktivieren. Wenn beispielsweise eine bestimmte Director-Administratorengruppe Benutzer nicht spiegeln soll, wählen Sie eine Rolle aus oder passen Sie eine Rolle an, die eine Spiegelung bzw. die benutzerdefinierte Berechtigung "Remoteunterstützung für Maschine ausführen" nicht zulässt, und die Schaltfläche Spiegeln wird nicht in der Benutzeroberfläche angezeigt.

Konfigurieren von benutzerdefinierten Rollen für Director-Administratoren

In Studio können Sie auch Director-spezifische benutzerdefinierte Rollen konfigurieren, die den Anforderungen Ihrer Organisation besser gerecht werden und eine flexiblere Delegation von Berechtigungen ermöglichen. Sie können beispielsweise die integrierte Helpdeskadministratorrolle einschränken, sodass dieser Administrator keine Sitzungen abmelden kann.

Wenn Sie eine benutzerdefinierte Rolle mit Director-Berechtigungen erstellen, müssen Sie dieser auch andere allgemeine Berechtigungen erteilen:

- Delivery Controller-Berechtigung zur Anmeldung bei Director.
- Berechtigungen für Bereitstellungsgruppen zum Anzeigen der zu diesen gehörigen Daten in Director.

Alternativ können Sie eine benutzerdefinierte Rolle erstellen, indem Sie eine vorhandene Rolle kopieren und dieser zusätzliche Berechtigungen für die verschiedenen Ansichten erteilen. Sie können beispielsweise die Helpdeskrolle kopieren und Berechtigungen zum Anzeigen des Dashboards oder der Ansicht "Filter" hinzufügen.

Legen Sie die Director-Berechtigungen für die benutzerdefinierte Rolle fest, einschließlich:

- Abbrechen von auf Maschine ausgeführter Anwendung erzwingen
- Abbrechen von auf Maschine ausgeführtem Prozess erzwingen
- Remoteunterstützung für Maschine ausführen

- Vorgang zum Zurücksetzen von vDisk ausführen
- Benutzerprofile zurücksetzen
- Anzeigen der Seite "Dashboard"
- Helpdeskseite anzeigen
- Hostverbindungsstatus und -benachrichtigungen anzeigen
- Slice-and-Dice-Seite anzeigen (Filteransicht)
- Trends anzeigen
- Anzeigen der Seite "Benutzerdetails"

Aus der Liste der Berechtigungen für andere Komponenten sollten Sie zusätzlich folgende Berechtigungen berücksichtigen:

- Profilverwaltung: Profildefinitionen lesen
- Bereitstellungsgruppen:
 - Aktivieren/Deaktivieren des Wartungsmodus einer Maschine mit der Bereitstellungsgruppenmitgliedschaft
 - Ausführen von Energievorgängen auf Windows-Desktopmaschinen mit der Bereitstellungsgruppenmitgliedschaft
 - Ausführen der Sitzungsverwaltung auf Maschinen unter mit der Bereitstellungsgruppenmitgliedschaft
 - Anzeigen von Bereitstellungsgruppen

Hinweis: Die Verfügbarkeit dieser Funktion richtet sich nach der Lizenzierung der Organisation und den Administratorberechtigungen.

HDX Insight ist die Integration der EdgeSight-Netzwerkanalyse und der EdgeSight-Leistungsverwaltung in Director:

- Die EdgeSight-Netzwerkanalyse nutzt HDX Insight und stellt eine kontextbezogene Ansicht der Anwendungen und Desktops im Netzwerk bereit. Mit diesem Feature bietet Director eine erweiterte Analyse des ICA-Datenverkehrs in der Bereitstellung.
- Die EdgeSight-Leistungsverwaltung bietet eine Verlaufsspeicherung und Trendberichte. Anhand der Beibehaltung historischer Daten können Sie im Gegensatz zur Echtzeitbewertung Trendberichte über Kapazität und Integrität usw. erstellen.

Nachdem Sie dieses Feature in Director aktivieren, liefert HDX Insight zusätzliche Informationen an Director:

- Auf der Seite Trends werden bereitstellungsübergreifend Auswirkungen auf Latenz und Bandbreite für Anwendungen, Desktops und Benutzer angezeigt.
- Auf der Seite Benutzerdetails werden Latenz- und Bandbreiteninformationen zu spezifischen Benutzersitzungen angezeigt.

Einschränkungen

- Die Roundtripzeitdaten für die ICA-Sitzung werden in Receiver für Windows 3.4 oder höher und Receiver für Mac 11.8 oder höher richtig angezeigt. Bei früheren Versionen von Receiver werden die Daten nicht richtig angezeigt.
- In der Trendansicht werden Anmeldedaten für HDX-Verbindungen für VDAs vor Version 7 nicht gesammelt. Für frühere VDAs werden die Diagrammdaten als 0 angezeigt.

Konfigurieren der EdgeSight-Netzwerkanalyse in Director

EdgeSight bietet eine Netzwerkanalyse unter Nutzung von NetScaler HDX Insight, anhand derer Administratoren der Citrix Anwendungen und Desktops Probleme behandeln und korrelieren können, die auf eine schlechte Netzwerkleistung zurückzuführen sind.

NetScaler Insight Center muss in Director installiert und konfiguriert sein, damit die EdgeSight-Netzwerkanalyse aktiviert werden kann. Insight Center ist eine virtuelle Maschine (Gerät), die von Citrix.com heruntergeladen wird. Mit der EdgeSight-Netzwerkanalyse sammelt Director Daten zur Bereitstellung. Diese Informationen werden aus HDX Insight verwendet, das eine stabile Analyse des Citrix ICA-Protokolls zwischen dem Client und der Citrix Back-End-Infrastruktur bietet.

1. Suchen Sie auf dem Server, auf dem Director installiert ist, das Befehlszeilentool DirectorConfig in `C:\inetpub\wwwroot\Director\tools` und führen Sie es mit dem Parameter `/confignetscaler` an der Eingabeaufforderung aus.
2. Wenn Sie dazu aufgefordert werden, konfigurieren Sie den Maschinennamen (FQDN oder IP-Adresse) für NetScaler Insight Center sowie den Benutzernamen, das Kennwort und den HTTP- oder HTTPS-Verbindungstyp.
3. Melden Sie sich zum Prüfen der Änderungen ab und wieder an.

Wenn Benutzer auf ihren Geräten VDAs einer Version vor XenDesktop 7 installiert haben, ergänzt Director Informationen aus der Bereitstellung durch Echtzeitstatus und -metrik über Windows-Remoteverwaltung (WinRM).

Darüber hinaus können Sie mit dem hier beschriebenen Verfahren WinRM für die Verwendung mit Remote PC in XenDesktop 5.6 Feature Pack1 konfigurieren.

Standardmäßig verfügen nur lokale Administratoren des Desktopcomputers (in der Regel Domänenadministratoren und andere privilegierte Benutzer) über die Berechtigungen, die zum Anzeigen der Echtzeitdaten erforderlich sind.

Informationen zum Installieren und Konfigurieren von WinRM finden Sie unter [CTX125243](#).

Um anderen Benutzern das Anzeigen von Echtzeitdaten zu ermöglichen, müssen Sie diesen Berechtigungen erteilen. Beispiel: Angenommen, es gibt mehrere Director-Benutzer (HelpDeskUserA, HelpDeskUserB usw.), die zu einer Active Directory-Sicherheitsgruppe namens "HelpDeskUsers" gehören. Der Gruppe wurde in Studio die Helpdeskadministratorrolle mit den erforderlichen Delivery Controller-Berechtigungen zugewiesen. Die Gruppe benötigt jedoch auch Zugriff auf Informationen vom Desktopcomputer.

Sie haben zwei Möglichkeiten, die Berechtigungen für den erforderlichen Zugriff zu konfigurieren:

- Erteilen von Berechtigungen für die Director-Benutzer (Identitätswechselmodell)
- Erteilen von Berechtigungen für den Director-Dienst (Modell mit vertrauenswürdigem Subsystem)

Erteilen von Berechtigungen für die Director-Benutzer (Identitätswechselmodell)

Director verwendet standardmäßig ein Identitätswechselmodell: Die WinRM-Verbindung mit der Desktopmaschine wird mit der Identität des Director-Benutzers hergestellt. Aus diesem Grund muss der Benutzer über die erforderlichen Berechtigungen auf dem Desktop verfügen.

Sie können diese Berechtigungen mit einer der folgenden beiden Methoden konfigurieren (später in diesem Abschnitt beschrieben):

1. Hinzufügen von Benutzern zur lokalen Administratorgruppe auf der Desktopmaschine.
2. Erteilen der von Director benötigten spezifischen Berechtigungen für Benutzer. Bei dieser Option wird vermieden, dass Director-Benutzer (beispielsweise die HelpDeskUsers-Gruppe) Volladministratorrechte auf der Maschine erhalten.

Erteilen von Berechtigungen für den Director-Dienst (Modell mit vertrauenswürdigem Subsystem)

Anstatt den Director-Benutzern Berechtigungen für die Desktopmaschinen zu erteilen, können Sie Director so konfigurieren, dass WinRM-Verbindungen mit einer Dienstidentität hergestellt werden, und ausschließlich dieser

Dienstidentität die entsprechenden Berechtigungen erteilen.

Bei diesem Modell sind die Benutzer von Director nicht berechtigt, selbst WinRM-Aufrufe zu tätigen. Sie können nur mit Director auf die Daten zugreifen.

Der Director-Anwendungspool in IIS ist für die Ausführung als Dienstidentität konfiguriert. Dies ist standardmäßig das virtuelle Konto "APPPOOL\Director". Beim Herstellen von Remoteverbindungen wird dieses Konto als das Active Directory-Computerkonto des Servers angezeigt, z. B. MyDomain\DirectorServer\$. Sie müssen dieses Konto mit den entsprechenden Berechtigungen konfigurieren.

Wenn mehrere Director-Websites bereitgestellt werden, müssen Sie das Computerkonto jedes Webserver in eine Active Directory-Sicherheitsgruppe setzen, die entsprechende Berechtigungen hat.

Um Director so einzustellen, dass anstelle der Benutzeridentität die Dienstidentität für WinRM verwendet wird, konfigurieren Sie die folgende Einstellung, wie unter [Erweiterte Konfiguration](#) beschrieben:

Service.Connector.WinRM.Identity = Service

Sie haben zwei Möglichkeiten, diese Berechtigungen zu konfigurieren:

1. Hinzufügen des Dienstkontos zur lokalen Administratorgruppe auf der Desktopmaschine.
2. Erteilen Sie dem Dienstkonto die für Director erforderlichen Berechtigungen (siehe weiter unten). So kann eine Erteilung von Volladministratorrechten für das Dienstkonto auf der Maschine vermieden werden.

Zuweisen von Berechtigungen für einen bestimmten Benutzer oder eine bestimmte Gruppe

Die folgenden Berechtigungen sind erforderlich, damit Director auf die von der Desktopmaschine benötigten Informationen über WinRM zugreifen kann:

- Lese- und Ausführberechtigungen in WinRM RootSDDL
- WMI-Namespace-Berechtigungen:
 - root/cimv2 – Remotezugriff
 - root/citrix – Remotezugriff
 - root/RSOP – Remotezugriff und Ausführen
- Zugehörigkeit zu diesen lokalen Gruppen:
 - Systemmonitorbenutzer
 - Ereignisprotokollleser

Das ConfigRemoteMgmt.exe-Tool, das zum automatischen Erteilen dieser Berechtigungen verwendet wird, befindet sich auf dem Installationsmedium in den Ordnern x86\Virtual Desktop Agent und x64\Virtual Desktop Agent und auf dem Installationsmedium im Ordner "tools". Sie müssen allen Director-Benutzern Berechtigungen erteilen.

Um einer Active Directory-Sicherheitsgruppe, einem Benutzer oder einem Computerkonto Berechtigungen zu erteilen, führen Sie das Tool mit Administratorrechten an einer Eingabeaufforderung mit den folgenden Argumenten aus:

ConfigRemoteMgmt.exe /configwinrmuser domain\name

wobei name eine Sicherheitsgruppe, ein Benutzer oder ein Computerkonto ist.

Beispiel: So erteilen Sie einer Benutzersicherheitsgruppe die erforderlichen Berechtigungen

ConfigRemoteMgmt.exe /configwinrmuser MyDomain\HelpDeskUsers

So erteilen Sie einem bestimmten Computerkonto Berechtigungen

ConfigRemoteMgmt.exe /configwinrmuser MyDomain\DirectorServer\$

Einige Elemente der erweiterten Director-Konfigurationen, z. B. die Unterstützung mehrerer Sites oder mehrerer Active Directory-Gesamtstrukturen, wird durch die Einstellungen im Internetinformationsdienste-Manager (IIS) gesteuert.

Wichtig: Wenn Sie eine Einstellung in IIS ändern, wird der Director-Dienst automatisch neu gestartet und die Benutzer werden abgemeldet.

Konfigurieren von erweiterten Einstellungen mit IIS

1. Öffnen Sie die IIS-Verwaltungskonsole.
2. Wechseln Sie zur Director-Website unter der Standardwebsite.
3. Doppelklicken Sie auf **Anwendungseinstellungen**.
4. Doppelklicken Sie auf eine Einstellung, um diese zu bearbeiten.

Unterstützten von Benutzern in mehreren Active Directory-Domänen und -Gesamtstrukturen

Director sucht in Active Directory nach Benutzern und nach zusätzlichen Benutzer- und Maschineninformationen. Standardmäßig durchsucht Director die folgende Domäne oder Gesamtstruktur:

- In der das Konto des Administrators Mitglied ist
- In der der Director-Webserver Mitglied ist (falls unterschiedlich)

Director versucht, Suchen auf Gesamtstrukturebene mit dem globalen Active Directory-Katalog durchzuführen. Wenn der Administrator keine Berechtigungen zum Suchen auf der Gesamtstrukturebene hat, wird nur die Domäne durchsucht.

Für die Suche nach Daten aus einer anderen Active Directory-Domäne oder Gesamtstrukturebene müssen Sie explizit die zu durchsuchenden Domänen oder Gesamtstrukturen festlegen. Konfigurieren Sie die folgenden Einstellungen:

Connector.ActiveDirectory.Domains = (Benutzer),(Server)

Die Werte der Attribute Benutzer und Server stellen die Domänen des Director-Benutzers (Administrator) bzw. des Director-Servers dar.

Um Suchen von einer zusätzlichen Domäne oder Gesamtstruktur zu ermöglichen, fügen Sie, wie in diesem Beispiel gezeigt, den Namen der Domäne der Liste hinzu:

Connector.ActiveDirectory.Domains = (Benutzer),(Server),ENDUSERDOMAIN

Director versucht, Suchen für jede Domäne in der Liste auf der Gesamtstrukturebene durchzuführen. Wenn der Administrator keine Berechtigungen zum Suchen auf der Gesamtstrukturebene hat, wird nur die Domäne durchsucht.

Hinzufügen von Sites zu Director

Wenn Director bereits installiert ist, richten Sie das Programm für die Arbeit mit mehreren Sites ein. Verwenden Sie hierzu die IIS-Manager-Konsole auf jedem Director-Server zum Aktualisieren der Liste der Serveradressen in den Anwendungseinstellungen.

Fügen Sie folgender Einstellung eine Controller-Adresse von jedem Standort hinzu:

Service.AutoDiscoveryAddresses = SiteAController,SiteBController

wobei SiteAController und SiteBController die Adressen von XenDesktop-Controllern zweier verschiedener Sites sind.

Deaktivieren der Sichtbarkeit von ausgeführten Anwendungen im

Aktivitätsmanager

Standardmäßig wird im Aktivitäts-Manager von Director eine Liste aller ausgeführten Anwendungen und die Windows-Beschreibung in den Titelleisten aller geöffneten Anwendungen für die Benutzersitzung angezeigt. Diese Informationen können von allen Administratoren angezeigt werden, die Zugriff auf den Aktivitätsmanager in Director haben. Bei delegierten Administratorrollen sind dies Volladministratoren, Bereitstellungsgruppenadministratoren und Helpdeskadministratoren.

Zum Schutz der Privatsphäre im Hinblick auf Benutzer und die von ihnen ausgeführten Anwendungen können Sie die Auflistung der ausgeführten Anwendungen auf der Registerkarte "Anwendungen" deaktivieren.

Achtung: Eine unsachgemäße Bearbeitung der Registrierung kann schwerwiegende Probleme verursachen und eine Neuinstallation des Betriebssystems erforderlich machen. Citrix übernimmt keine Garantie dafür, dass Probleme, die auf eine unsachgemäße Verwendung des Registrierungs-Editors zurückzuführen sind, behoben werden können. Die Verwendung des Registrierungs-Editors geschieht daher auf eigene Gefahr. Sichern Sie die Registrierung auf jeden Fall vor dem Bearbeiten ab.

1. Ändern Sie für den VDA den Registrierungsschlüssel in HKLM\Software\Citrix\Director\TaskManagerDataDisplayed. Standardmäßig ist dieser Schlüssel auf 1 eingestellt. Ändern Sie den Wert auf 0, was bedeutet, dass die Informationen nicht im Aktivitätsmanager angezeigt werden.
2. Bearbeiten Sie auf dem Server, auf dem Director installiert ist, die Einstellung zur Steuerung der Sichtbarkeit ausgeführter Anwendungen. In der Standardeinstellung ist der Wert "Wahr", wodurch die Sichtbarkeit der ausgeführten Anwendungen auf der Registerkarte Anwendungen zugelassen wird. Ändern Sie den Wert in "Falsch", wodurch die Sichtbarkeit deaktiviert wird. Diese Option gilt nur für den Aktivitätsmanager in Director, nicht für den VDA.
Ändern Sie den Wert der folgenden Einstellung:
`UI.TaskManager.EnableApplications = false`

Wichtig: Zum Deaktivieren der Ansicht ausgeführter Anwendungen empfiehlt Citrix, dass beide Änderungen durchgeführt werden, damit die Daten im Aktivitätsmanager nicht angezeigt werden.

Behandeln von Benutzerproblemen

Sep 29, 2015

Im Aktivitäts-Manager in Director zeigen Sie Informationen über den Benutzer an:

- Überprüfen Sie die Details zur Anmeldung des Benutzers, zur Verbindung und zu den Anwendungen.
- Spiegeln Sie die Maschine des Benutzers.
- Behandeln Sie das Problem mit den in der folgenden Tabelle empfohlenen Aktionen und eskalieren Sie das Problem ggf. an den entsprechenden Administrator.

Problem des Benutzers	Lösungsvorschläge:
Anmeldung dauert lange oder schlägt periodisch oder wiederholt fehl	Diagnose von Benutzeranmeldeproblemen
Anwendung ist langsam oder reagiert nicht mehr	Beheben von Anwendungsfehlern
Verbindung ist fehlgeschlagen	Wiederherstellen von Desktopverbindungen und -sitzungen
Sitzung ist langsam oder reagiert nicht	Wiederherstellen von Sitzungen
Video ist langsam oder von schlechter Qualität	Ausführen von Systemberichten für HDX-Kanäle

Hinweis: Um sicherzustellen, dass der Computer nicht im Wartungsmodus ist, überprüfen Sie in der Ansicht Benutzerdetails den Bereich Maschinendetails.

In der Ansicht "Benutzerdetails" werden im Bereich "HDX" keine zugeordneten Clientlaufwerke angezeigt und zugeordnete Drucker werden nicht im Druckkanal erkannt.

Führen Sie Citrix Scout auf einem Delivery Controller oder VDA aus, um wichtige Datenpunkte und CDF-Traces (Citrix Diagnosis Facility) für die Fehlerbehebung auf ausgewählten Computern zu erfassen. Nach der Erfassung dieser Informationen lädt Scout die Datenpunkte sicher zum technischen Support von Citrix hoch. Die TaaS- (Tools-as-a-Service-)Plattform reduziert anhand dieser Informationen die Zeit zur Lösung der von Kunden gemeldeten Probleme.

Scout wird mit XenApp- bzw. XenDesktop-Komponenten installiert. Scout wird im Windows-Startmenü oder im Startbildschirm von Windows 8 oder 8.1 angezeigt, wenn Sie XenDesktop 7.1, XenDesktop 7.5 oder XenApp 7.5 installieren oder ein Upgrade darauf durchführen.

Zum Starten von Scout über das Startmenü oder den Startbildschirm wählen Sie Citrix > Citrix Scout.

Weitere Informationen zum Verwenden und Konfigurieren von Scout und häufig gestellte Fragen finden Sie unter <http://support.citrix.com/article/CTX130147>.

Im folgenden Video wird die Verwendung von Scout kurz erläutert.

Wenn Sie den Namen des Benutzers im Suchfeld eingeben, sucht Director in Active Directory nach Benutzern in allen Sites, die für Director konfiguriert wurden.

Die Suchergebnisse schließen auch Benutzer ein, die derzeit keine Maschine verwenden bzw. keiner Maschine zugewiesen sind.

- Bei der Suche wird die Groß- und Kleinschreibung nicht beachtet.
- Teileinträge ergeben eine Liste möglicher Übereinstimmungen.
- Nachdem Sie einige Buchstaben eines zweiteiligen Namens (Benutzername, Nachname und Vorname oder Anzeigename) durch Leerzeichen getrennt eingegeben haben, enthalten die Ergebnisse Übereinstimmungen für beide Zeichenfolgen. Wenn Sie beispielsweise `jo robe` eingeben, werden Zeichenfolgen wie "John Robertson" oder "Robert, Jones" als Ergebnisse angezeigt.

Klicken Sie auf das Director-Logo, um zur Startseite zurückzukehren.

Überwachen von Bereitstellungen

Sep 29, 2015

Wenn Sie Director mit Volladministratorrechten öffnen, erscheint das Dashboard zur Überwachung der Integrität und Nutzung einer Site.

Wenn es zurzeit keine Fehler gibt und keine Fehler in den letzten 60 Minuten aufgetreten sind, bleiben Bereiche ausgeblendet. Wenn Fehler auftreten, wird der zugehörige Fehlerbereich automatisch angezeigt.

Hinweis: Je nachdem, über welche Lizenzierung Ihre Organisation verfügt und welche Administratorrechte vorliegen, stehen einige Optionen oder Features möglicherweise nicht zur Verfügung.

Bereich	Beschreibung
Benutzerverbindungsfehler	Verbindungsfehler während der letzten 60 Minuten. Klicken Sie auf die Kategorien neben der Gesamtzahl zum Anzeigen von Metriken für diesen Fehlertyp. In der nebenstehenden Tabelle wird angezeigt, wie sich dieser Wert auf die Bereitstellungsgruppen verteilt.
Fehlgeschlagene Desktopbetriebssystemmaschinen bzw. Fehlgeschlagene Serverbetriebssystemmaschinen	Gesamtanzahl der Fehler in den letzten 60 Minuten unterteilt nach Bereitstellungsgruppen. Fehler unterteilt nach Typ, einschließlich "konnte nicht gestartet werden", "beim Starten hängen geblieben" und "nicht registriert". Bei Serverbetriebssystemmaschinen wird auch das Erreichen der maximalen Last angegeben.
Verbundene Sitzungen	Verbunden Sitzungen in allen Bereitstellungsgruppen in den letzten 60 Minuten.
Durchschnittliche Anmeldedauer	Anmeldedaten für die letzten 60 Minuten. Die große Zahl links ist die durchschnittliche Anmeldedauer während einer Stunde. Anmeldedaten für VDAs vor XenDesktop 7.0 sind nicht in diesem Durchschnitt enthalten.
Infrastruktur	Integritätsstatus der Hosts, Controller und Infrastruktur. Anzeige von Leistungswarnungen. Für Hosts werden der Verbindungsstatus sowie der Zustand der CPU, Arbeitsspeicher, Bandbreite (Netzwerkverwendung) und Speicher (Datenträgernutzung) anhand von Informationen von XenServer oder VMware überwacht. Sie können beispielsweise XenCenter so konfigurieren, dass Warnungen zur Leistung generiert werden, wenn die CPU-, Netzwerk-E/A- oder Datenträger-E/A-Nutzung einen angegebenen Schwellenwert auf einem verwalteten Server oder einer virtuellen Maschine übersteigt. Standardmäßig ist das Warnungswiederholungsintervall 60 Minuten, Sie können jedoch auch eine andere Einstellung wählen. Weitere Informationen finden Sie in der XenServer-Dokumentation unter Konfigurieren von Leistungswarnungen.

Hinweis: Wird für eine bestimmte Metrik kein Symbol angezeigt, bedeutet dies, dass diese Metrik von dem verwendeten Hosttyp nicht unterstützt wird. Beispiel: Für System Center Virtual Machine Manager-Hosts (SCVMM) sind keine Integritätsdaten verfügbar.

Fahren Sie zum Beheben von Problemen mit den folgenden Optionen fort:

- [Steuern der Energieverwaltung der Benutzermaschine](#)
- [Überwachen von Sitzungen und Verhindern von Verbindungen](#)

Überwachen von Personal vDisks

Sep 29, 2015

Sie können ein Diagnosetool zum Überwachen der Änderungen verwenden, die von Benutzern an beiden Bereichen persönlicher vDisks (Benutzerdaten- und Anwendungsbereiche) vorgenommen werden. Zu diesen Änderungen gehören Anwendungen, die von Benutzern installiert wurden, und von ihnen geänderte Dateien. Die Änderungen werden in einer Reihe von Berichten erfasst.

1. Führen Sie auf der zu überwachenden Maschine C:\Programme\Citrix\Personal vDisk\bin\CtxPvdDiag.exe aus.
2. Navigieren Sie zu einem Speicherort, an dem die Berichte und Protokolle gespeichert werden sollen, legen Sie fest, welche Berichte generiert werden sollen, und klicken Sie auf OK. Die folgenden Berichte sind verfügbar:

Bericht oder Protokoll	Generierte Dateien	Überwachte Änderungen
Softwarestrukturbericht	Software.Dat.Report.txt, Software.Dat.delta.txt	Software.Dat.Report.txt erfasst Änderungen, die vom Benutzer an der Struktur HKEY_LOCAL_MACHINE\Software vorgenommen werden. Der Bericht besteht aus den folgenden Abschnitten: • List of Applications installed on the base: Anwendungen, die in Ebene 0 installiert wurden. • List of user installed software: Anwendungen, die vom Benutzer im Anwendungsbereich der vDisk installiert wurden. • List of software uninstalled by user: Ursprünglich in Ebene 0 installierte Anwendungen, die vom Benutzer entfernt wurden. Weitere Informationen zu Software.Dat.delta.txt finden Sie im Strukturdeltabericht.
Systemstrukturbericht	SYSTEM.CurrentControlSet.DAT.Report.txt	In dieser Datei werden die Änderungen erfasst, die vom Benutzer an der Struktur HKEY_LOCAL_MACHINE\System vorgenommen werden. Der Bericht besteht aus den folgenden Abschnitten: • List of user installed services: Vom Benutzer installierte Dienste und Treiber. • Startup of following services were changed: Dienste und Treiber, deren Starttyp vom Benutzer geändert wurde.
Sicherheitsstrukturbericht	SECURITY.DAT.Report.txt	Diese Datei dient zur Überwachung aller Änderungen, die der Benutzer in der

Bericht oder Protokoll	Generierte Dateien	Struktur HKEY_LOCAL_MACHINE\Security Überwachte Änderungen vornimmt.
Strukturbericht für die Sicherheitskontenverwaltung (SAM)	SAM.DAT.Report.txt	Diese Datei dient zur Überwachung aller Änderungen, die der Benutzer in der Struktur HKEY_LOCAL_MACHINE\SAM vornimmt.
Strukturdeltabericht	Software.Dat.delta.txt	In dieser Datei werden alle Registrierungsschlüssel und Werte erfasst, die vom Benutzer in der Struktur HKEY_LOCAL_MACHINE\Software hinzugefügt oder entfernt wurden, sowie alle geänderten Werte.
Personal vDisk-Protokolle	Pvd-IvmSupervisor.log, PvdActivation.log, PvdSvc.log, PvdWMI.log, SysVol-IvmSupervisor.log, vDeskService-<#>.log	Diese Dateien werden standardmäßig in P:\Users\Local\Temp\PVDLOGS generiert und dann aber an den ausgewählten Speicherort verschoben.
Windows-Betriebssystemprotokoll	EvtLog_App.xml, EvtLog_System.xml, setupapi.app.log, setuperr.log, setupapi.dev.log, msinfo.txt	EvtLog_App.xml und EvtLog_System.xml sind die Anwendungs- und Systemereignisprotokolle im XML-Format aus dem persönlichen vDisk-Volumen. Die Protokolle Setupapi.app.log und setuperr.log enthalten Protokollmeldungen bezüglich der Ausführung von msixec.exe bei der Installation von Personal vDisk. Setupapi.dev.log enthält Protokollmeldungen über die Geräteinstallation. Msinfo.txt enthält die Ausgabe von msinfo32.exe. Informationen zu dieser Ausgabe finden Sie in der Microsoft-Dokumentation.
Dateisystembericht	FileSystemReport.txt	In dieser Datei werden die Änderungen erfasst, die vom Benutzer am Dateisystem vorgenommen werden. Der Bericht besteht aus den folgenden Abschnitten: • Files Relocated: Dateien in Ebene 0, die vom Benutzer zur vDisk verschoben wurden. Dateien der Ebene 0 sind Dateien, die aus dem Masterimage von der Maschine geerbt wurden, der die persönliche vDisk zugeordnet ist. • Files Removed: Dateien in Ebene 0, die

Bericht oder Protokoll	Generierte Dateien	Überwachte Änderungen
		durch eine Benutzeraktion (z. B. Entfernen einer Anwendung) verborgen wurden. • Files Added (MOF,INF,SYS): Dateien mit der Erweiterung "mof", "inf" oder "sys", die vom Benutzer der vDisk hinzugefügt wurden (z. B. bei der Installation einer Anwendung wie Visual Studio 2010, bei der eine MOF-Datei für die automatische Wiederherstellung registriert wird). • Files Added Other: Andere Dateien, die der vDisk vom Benutzer hinzugefügt wurden (z. B. beim Installieren einer Anwendung). • Base Files Modified But Not Relocated: Dateien in Ebene 0, die vom Benutzer geändert wurden, jedoch nicht von den Personal vDisk-Kernelmodustreibern in der vDisk erfasst wurden.

Verwenden der Überwachungsdienst-OData-API

Sep 29, 2015

In diesem Abschnitt wird die Anwendungsprogrammierschnittstelle (Application Programming Interface, API) des Überwachungsdiensts vorgestellt und beschrieben, welche Daten über die API zur Verfügung stehen und wie darauf zugegriffen wird. Der Abschnitt enthält auch Beispiele zur Verwendung der API. Außerdem wird erläutert, wie die Endpunkte der Überwachungsdienst-OData-API mit SSL geschützt werden. Diese Informationen richten sich an erfahrene Administratoren, die mit der Arbeit mit OData-APIs vertraut sind.

Zusätzlich zur Anzeige von historischen Daten in der Citrix Director-Konsole können Sie Daten mit der Überwachungsdienst-OData-API abfragen. Die API verwendet ein umfassendes Framework, das große Flexibilität für die Abfrage von Daten bietet. Sie können die API zum Analysieren historischer Trends für die Planung oder für die gezielte Problembehandlung bei Verbindungs- und Maschinenfehlern einsetzen. Mit der API können zudem Informationen für die Eingabe in andere Tools und Prozesse extrahiert werden, z. B. wenn PowerPivot-Tabellen in Microsoft Excel für die Anzeige von Daten auf verschiedene Weise verwendet werden. Sie können auch eine benutzerdefinierte Benutzeroberfläche auf der Basis der von der API bereitgestellten Daten erstellen.

Die Überwachungsdienst-OData-API verwendet das Open Data-Protokoll (OData), ein Webprotokoll für die Abfrage und Aktualisierung von Daten, das auf Webtechnologien wie HTTP aufbaut. Weitere Informationen über das OData-Protokoll finden Sie unter <http://www.odata.org>.

Über die API verfügbare Daten

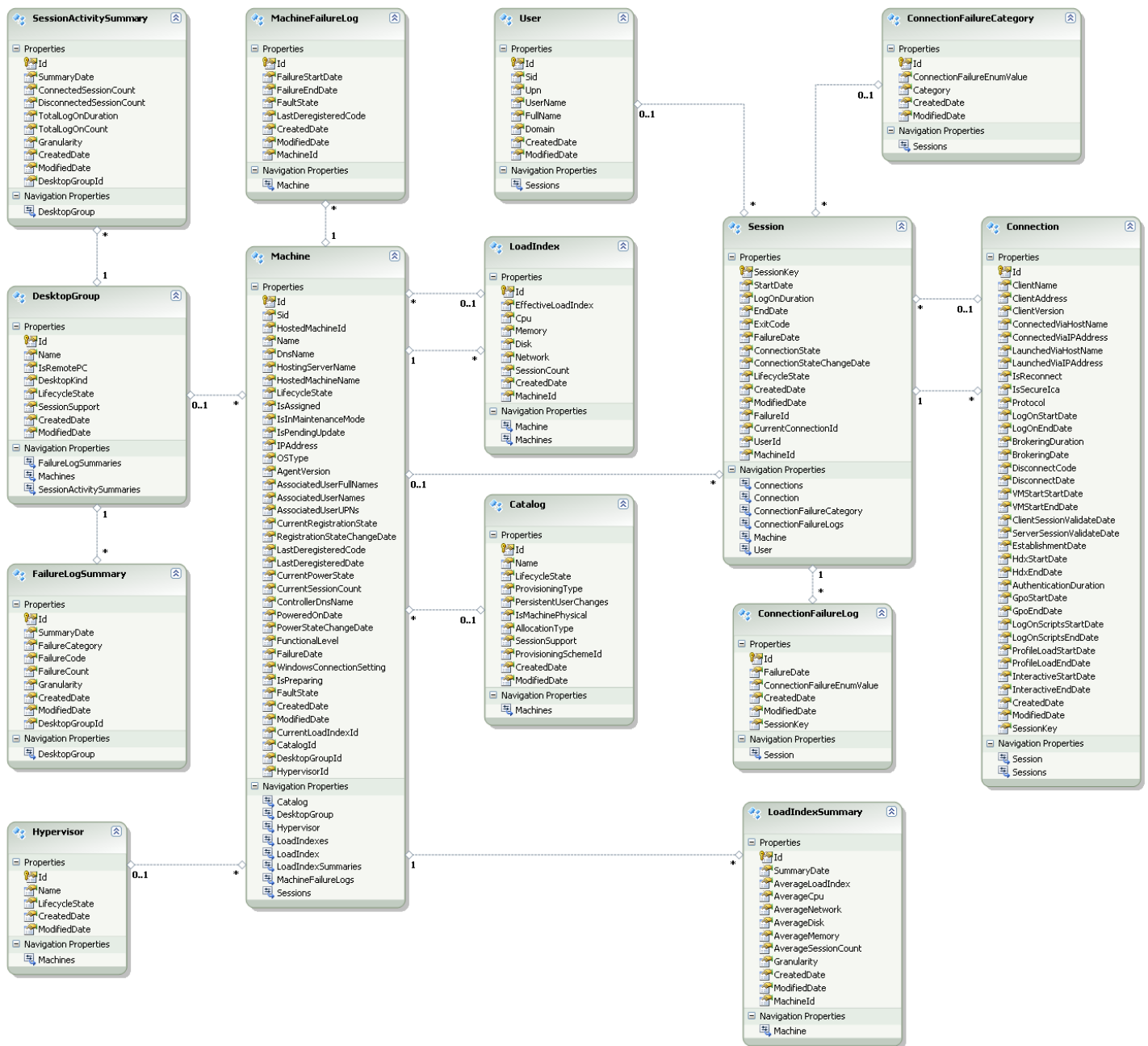
Sep 29, 2015

In diesem Abschnitt werden die Daten beschrieben, auf die mit der Überwachungsdienst-OData-API zugegriffen werden kann. Er enthält zudem eine Schemaabbildung.

Das Schema des Überwachungsdiensts bietet folgende Datentypen:

- Daten zu Verbindungsfehlern
- Maschinen in fehlerhaftem Zustand
- Sitzungsnutzung
- Anmeldedauer
- Daten zum Lastausgleich

Die folgende Abbildung zeigt das Überwachungsdienst-Schema.



Zugriff auf Daten mit der API

Sep 29, 2015

Im folgenden Abschnitt wird erläutert, wie Sie auf Daten mit der Überwachungsdienst-OData-API zugreifen.

Die Überwachungsdienst-API setzt mit den Windows Communication Foundation (WCF) Data Services auf SQL Server-Datenbanken auf, die während der Verarbeitung und Konsolidierung aufgefüllt werden. Bei Verwendung von WCF mit wsHttpBinding werden zwei Endpunkte verfügbar gemacht. Die Basisadresse ist `http://{dc-host}/Citrix/Monitor/OData/v1`. Sie können Endpunkte auch mit SSL schützen (weitere Informationen finden Sie unter [Schützen von Endpunkten mit SSL](#)).

1. Der Endpunkt "Methods" macht Dienstvorgänge verfügbar, die von Citrix Director für das Abrufen von Daten verwendet werden, welche eine komplexe Gruppierung und hohe Leistungsstandards erfordern (z. B. Abfragen für das Dashboard und Trend-Seiten). Die URI der Methods-API ist: `http://{dc-host}/Citrix/Monitor/OData/v1/Methods`.
2. Der Endpunkt "Data" gestattet direkten Lesezugriff auf die Datenbankentitäten, Zugriff ist mit der Abfragesprache OData möglich. Dieser Endpunkt gestattet einen sehr flexiblen Zugriff im Hinblick auf Filter und Spaltenauswahl, bietet aber nicht die gleichen Leistungsvorteile, die mit hochspezifischen Dienstvorgängen einhergehen. Die URI der Data-API ist: `http://{dc-host}/Citrix/Monitor/OData/v1/Data`.

Authentifizierung

Zur Verwendung der Überwachungsdienst-OData-API müssen Sie XenApp- bzw. XenDesktop-Administrator sein. Zum Aufrufen der API ist Lesezugriff erforderlich. Welche Daten zurückgegeben werden, hängt von den Rollen und Berechtigungen des XenApp- bzw. XenDesktop-Administrators ab. Administratoren von Bereitstellungsgruppen können beispielsweise die Überwachungsdienst-API aufrufen, aber die zurückgegebenen Daten hängen von dem mit Citrix Studio eingerichteten Bereitstellungsgruppenzugriff ab. Weitere Informationen über XenApp-/XenDesktop-Administratorrollen und -berechtigungen finden Sie unter [Delegierte Administration](#).

Abfragen von Daten

Die Überwachungsdienst-API ist eine REST-basierte API, auf die mit einem OData-Consumer zugegriffen werden kann. OData-Consumer sind Anwendungen, die mit dem OData-Protokoll verfügbar gemachte Daten nutzen. Die Ausführung von OData-Consumern reicht vom einfachen Webbrowser bis zur benutzerdefinierten Anwendung, die sämtliche Features des OData-Protokolls nutzt. Weitere Informationen über OData-Consumer finden Sie unter <http://www.odata.org/ecosystem#consumers>.

Jeder Teil des Überwachungsdienst-Datenmodells ist zugänglich und kann über die URL gefiltert werden. OData bietet eine Abfragesprache im URL-Format zum Abrufen von Einträgen aus einem Dienst. Weitere Informationen finden Sie unter <http://msdn.microsoft.com/en-us/library/ff478141.aspx>.

Die Abfrage wird serverseitig verarbeitet und kann clientseitig mit dem OData-Protokoll weiter gefiltert werden.

Hinweis: Enumerationen werden vom OData-Protokoll nicht unterstützt, stattdessen werden ganze Zahlen verwendet. Weitere Informationen zum Bestimmen der von der Überwachungsdienst-OData-API zurückgegebenen Werte finden Sie unter [Bestimmen der Enumerationswerte](#).

Datenkategorien

Der Überwachungsdienst erfasst diverse Daten über Benutzersitzungsnutzung, Benutzeranmeldeleistung, Sitzungslastausgleich und zu Fehlern bei Verbindungen und Maschinen. Die Daten werden je nach Kategorie unterschiedlich aggregiert. Zum Interpretieren der Daten sind Kenntnisse über die Aggregation der mit den OData-Methoden-APIs

abgerufenen Datenwerte unverzichtbar. Beispiel:

- Fehler bei verbundenen Sitzungen und Maschinen treten über einen bestimmten Zeitraum hinweg auf und werden daher als Maximalwerte über einen Zeitraum hinweg verfügbar gemacht.
- Die Anmeldedauer ist ein Zeitlängenwert und wird daher als Durchschnitt über einen Zeitraum hinweg verfügbar gemacht.
- Die Anzahl der Anmeldungen und Verbindungsfehler repräsentieren eine Anzahl von Vorkommen über einen bestimmten Zeitraum und werden als Summen über einen Zeitraum hinweg verfügbar gemacht.

Auswertung gleichzeitiger Daten

Sitzungen müssen sich überschneiden, um als gleichzeitig angesehen zu werden. Bei einem Zeitintervall von 1 Minute werden jedoch alle Sitzungen in dieser Minute (mit oder ohne Überschneidung) als gleichzeitig angesehen, d. h. das Intervall ist so klein, dass sich der Mehraufwand für die Berechnung der Genauigkeit nicht lohnt. Finden die Sitzungen in der gleichen Stunde, aber nicht in der gleichen Minute statt, werden sie als einander nicht überschneidend angesehen.

Informationen über die Zeit- und Datumsbereiche

Sep 29, 2015

In diesem Abschnitt wird erläutert, wie Datums- und Uhrzeitbereiche in der Überwachungsdienst-OData-API angegeben und interpretiert werden. Die Informationen in diesem Abschnitt müssen gelesen werden, wenn der Methods-Endpunkt verwendet wird.

Beim Festlegen von Zeitstempelbereichen in der Überwachungsdienst-OData-API ist zu beachten, dass solche Bereiche einschließlich sind. Das heißt, der Zeitstempel umfasst sämtliche Vorkommnisse während eines Intervalls bis zum Anfang des nächsten Intervalls. Angeben derselben Zeit für Beginn und Ende eines Zeitbereichs ergibt einen einzelnen Datenpunkt in der angeforderten Granularität.

Dieses Modell bietet eine konsistente Anwendung von Zeit- und Datumsbereichsangaben in einer stufenweisen Granularität von Minuten bis zu Monaten. Die folgende Tabelle zeigt einige Beispiele.

Bereich (Beginn-Ende)	Intervall/Granularität	Zurückgegebene Datenpunkte	Daten
1:00-1:00	1 Minute	1	deckt 1:00:00-1:00:59 ab
1:00-01:01	1 Minute	2	1:00, 1:01 und deckt 1:00:00-1:01:59 ab
1:00-02:00	1 Minute	61	einschl. bis 2:00:59
01:07-01:12	1 Minute	6	01:07-01:12:59
1:00-1:00	5 Minuten	1	1:00-01:04:59
01:05-01:10	5 Minuten	2	01:05-01:14:59
1:00-02:00	1 Stunde	2	einschl. 1:00-2:59
01.01.2012-02.01.12	1 Tag	2	1.1 und 1.2
01.01.2012-12.12.12	1 Monat	12	Daten jedes Monats

Schützen von Endpunkten mit SSL

Sep 29, 2015

In diesem Abschnitt wird erläutert, wie Sie die Endpunkte der Überwachungsdienst-OData-API mit SSL schützen. Wenn Sie SSL verwenden, müssen Sie SSL auf allen Delivery Controllern in der Site konfigurieren. Sie können keine Mischung aus Controllern mit und ohne SSL verwenden.

Zum Schützen der Endpunkte des Überwachungsdiensts mit SSL müssen Sie die folgende Konfiguration vornehmen. Einige Schritte müssen nur einmal pro Site vorgenommen werden, andere hingegen auf jeder Maschine, auf der der Überwachungsdienst gehostet wird. Die Schritte werden nachfolgend beschrieben.

Teil 1: Zertifikatregistrierung im System

1. Erstellen Sie mit einem vertrauenswürdigen Zertifikat-Manager ein Zertifikat. Das Zertifikat muss dem Port auf der Maschine zugeordnet werden, den Sie für OData SSL verwenden möchten.
2. Konfigurieren Sie den Überwachungsdienst zur Verwendung dieses Ports für SSL-Kommunikation. Die Schritte hängen von der jeweiligen Umgebung ab und davon, wie diese mit Zertifikaten funktioniert. Das folgende Beispiel zeigt das Konfigurieren von Port 449:
 - Ordnen Sie das Zertifikat einem Port zu:
`netsh http add sslcert ipport=0.0.0.0:449 certhash=97bb629e50d556c80528f4991721ad4f28fb74e9appid='{00000000-0000-0000-0000-000000000000}'`
Tipp: Stellen Sie in einem PowerShell-Befehlsfenster sicher, dass Sie die GUID in der Anwendungs-ID mit einfachen Anführungszeichen wie oben dargestellt einschließen, da der Befehl sonst nicht funktioniert. Beachten Sie, dass diesem Beispiel eine Zeile für bessere Lesbarkeit hinzugefügt wurde.

Teil 2: Ändern der Konfigurationseinstellungen des Überwachungsdiensts

1. Führen Sie auf jedem Delivery Controller in der Site folgende PowerShell-Befehle einmal aus. Damit wird die Registrierung des Überwachungsdiensts beim Konfigurationsdienst aufgehoben.
`asnp citrix.* $ServiceGroup = get-configregisteredServiceInstance -servicetype Monitor | Select -First 1 ServiceGroupUid remove-configServiceGroup -ServiceGroupUid $ServiceGroup.ServiceGroupUic`
2. Führen Sie den folgenden Schritt auf allen Controllern in der Site aus:
 - Verwenden Sie eine Eingabeaufforderung zum Suchen des Verzeichnisses des installierten Citrix Überwachungsdiensts (in der Regel C:\Programme\Citrix\Monitor\Service). Führen Sie in diesem Verzeichnis Folgendes aus:
`Citrix.Monitor.Exe -CONFIGUREFIREWALL -ODataPort 449 -RequireODataSsl`
 - Führen Sie die folgenden PowerShell-Befehle aus:
`asnp citrix.* (falls dies noch nicht in diesem Fenster ausgeführt wird) get-MonitorServiceInstance | register-ConfigServiceInstance Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-M`

Methoden

Sep 29, 2015

In den folgenden Abschnitten werden die von der API verfügbar gemachten Methoden beschrieben. Neben den Methoden werden die angegebenen Parameter und die vom API-Aufruf zurückgegebenen Elemente erläutert. Zudem werden Beispiele für Abfragen gegeben.

Weitere Informationen zur Angabe der Zeitintervalle finden Sie unter [Info über die Zeit und Datumsbereiche](#). Weitere Informationen zu den von der Überwachungsdienst-OData-API zurückgegebenen Werten siehe [Bestimmen der Enumerationswerte](#).

Ruft die durchschnittliche Anmeldedauer für den angegebenen Zeitraum, die Benutzer-ID und den Parameter "desktopGroupId" ab.

Parametername	Typ	Anmerkungen
startDate	DateTime	Beginn des Zeitraums, für den der Durchschnitt abgerufen wird. Dies muss eine UTC-Angabe sein.
endDate	DateTime	Ende des Zeitraums, für den der Durchschnitt abgerufen wird. Dies muss eine UTC-Angabe sein.
userSid	Zeichenfolge	Benutzer-SID, auf die der Durchschnitt beschränkt wird.
desktopGroupId	GUID	Desktopgruppen-UID, auf die der Durchschnitt beschränkt wird.

Zurückgegebene Elemente

Ein IQueryable von LogOnBreakdown-Objekten, die für jeden Anmeldungsschritt Folgendes enthalten:

Eigenschaft	Typ	Anmerkungen
LogonStepItems	List	Liste der LogOnStepItems. Jeder LogonStepItem besitzt einen LogonStep (mögliche Werte sind Brokering=1, VMStart=2, HDX=3, Authentication=4, Gpos=5, LogonScripts=6, ProfileLoad=7, Interactive=8, Total=0) und eine Duration (Nullable) in Millisekunden. Wenn keine Daten für einen Schritt verfügbar sind, wird die Dauer mit NULL zurückgegeben.
BreakdownType	int	Typ der Aufschlüsselung. UsersLastSession=1, UsersSessionAverage=2, DesktopGroupAverage=3

Hinweis: Die HDX-Dauer ist eine neue Metrik, die Änderungen am ICA-Protokoll erfordert. Das bedeutet, dass wenn die neue Version des Clients nicht verwendet wird, als Metriken NULL zurückgegeben wird.

Beispiel

Ruft die durchschnittliche Anmeldedauer am 15. März, für diese Benutzer-ID und desktopGroupId ab.

`http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetAverageLogOnBreakdown?startDate=datetime'2012-03-15T00:00:00'&endDate=datetime'2012-03-30T00:00:00'&userSid='User%20Sid'&desktopGro`

Ruft den Trend der durchschnittlichen Anmeldedauer für den angegebenen Zeitraum ab. Abgedeckt werden nur erfolgreiche Anmeldungen (d. h. die einen LogOnStartDate- und einen LogOnEndDate-Eintrag in der Datenbank haben).

Parametername	Typ	Anmerkungen
start	DateTime	Beginn des Zeitraums, für den der Trend der durchschnittlichen Anmeldedauer abgefragt wird. Dies muss eine UTC-Angabe sein.
end	DateTime	Ende des Zeitraums, für den der Trend der Verbindungsfehler abgefragt wird. Dies muss eine UTC-Angabe sein.

Zurückgegebene Elemente

Double, das die durchschnittliche Anmeldedauer für den angegebenen Zeitraum repräsentiert.

Beispiel

Ruft die durchschnittliche Anmeldedauer für eine Stunde für alle Bereitstellungsgruppen ab.

`http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetAverageLogOnDuration?start=datetime'2012-06-05T12:24:00'&end=datetime'2012-06-05T13:24:00'`

Ruft Datenpunkte für den angegebenen Zeitraum ab zur Darstellung eines Trends der gleichzeitigen Sitzungen in Form einer Liste von DateTime/Wert-Paaren. Jeder Punkt stellt die maximale Anzahl gleichzeitiger Sitzungen während des angegebenen Intervalls dar. Der letzte Punkt des Trends kann die Anzahl aktueller Benutzer im System darstellen.

Parametername	Typ	Anmerkungen
startDate	DateTime	Beginn des Zeitraums, für den die gleichzeitigen Sitzungen abgefragt werden. Dies muss eine UTC-Angabe sein. Inklusiv.
endDate	DateTime	Ende des Zeitraums, für den die gleichzeitigen Sitzungen abgefragt werden. Dies muss eine UTC-Angabe sein. Inklusiv.
intervallLength	int	Intervall zwischen Datenpunkten in Minuten für den Trend.

Parametername	Typ	Anmerkungen
---------------	-----	-------------

IQueryable-Liste von TrendItem-Objekten – IQueryable, aufsteigend nach Datum sortiert.

Jedes TrendItem-Objekt enthält eine UTC-Datumeigenschaft (DateTime) und eine Werteigenschaft (double) zum Zeichnen des Trends.

Beispiel

Ruft den Trend gleichzeitiger Sitzungen über eine Stunde ab (Datenpunkt pro Minute):

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetConcurrentSessionsTrend?startDate=datetime'2012-06-05T12:26:00'&endDate=datetime'2012-06-05T13:26:00'&intervalLength=1

Ruft den Trend verbundener Benutzer für Serverbetriebssystemmaschinen während des angegebenen Zeitraums ab. Jeder Datenpunkt des Trends repräsentiert die Gesamtzahl der mit Serverbetriebssystemmaschinen verbundenen Benutzer im letzten Intervall (intervalLength).

Parametername	Typ	Anmerkungen
startDate	DateTime	Beginn des Zeitraums, für den der Trend verbundener Benutzer abgefragt wird. Dies muss eine UTC-Angabe sein.
endDate	DateTime	Ende des Zeitraums, für den der Trend verbundener Benutzer abgefragt wird. Dies muss eine UTC-Angabe sein.
intervalLength	int	Intervall zwischen Datenpunkten in Minuten für den Trend.
desktopGroupsFilter	string	Durch Trennzeichen getrennte Liste der Bereitstellungsgruppen-UIDs, auf die das Ergebnis beschränkt wird.
machineFilter	string	Durch Trennzeichen getrennte Liste der Maschinen-SIDs, auf die das Ergebnis beschränkt wird.

Zurückgegebene Elemente

Ein IQueryable von Objekten mit folgendem Inhalt:

Eigenschaft	Typ	Anmerkungen
ConnectedUsersTrend	List	TrendItem enthält ein UTC-Datum (DateTime) und eine Anzahl (int) für die maximale Anzahl der verbundenen Benutzer im Intervall.

Beispiele

Ruft einen 24-Stunden-Trend (Datenpunkt alle 15 Minuten) der verbundenen Benutzer für die beiden angegebenen Bereitstellungsgruppen ab.

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetConnectedUsersTrend?startDate=datetime'2011-09-30T00:00:00'&endDate=datetime'2011-10-01T00:00:00'&intervalLength=15&desktopGroupsFilter=
Ruft einen Wochentrend (Datenpunkt alle 4 Stunden) der verbundenen Benutzer ab.

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetConnectedUsersTrend?startDate=datetime'2011-09-23T00:00:00'&endDate=datetime'2011-10-01T00:00:00'&intervalLength=240

Ruft den Trend der Verbindungsfehler für den angegebenen Zeitraum ab. Jeder Datenpunkt im Trend repräsentiert die Gesamtzahl der Verbindungsfehler im letzten Intervall (intervalLength-Parameter).

Parametername	Typ	Anmerkungen
startDate	DateTime	Beginn des Zeitraums, für den der Trend der Verbindungsfehler abgefragt wird. Dies muss eine UTC-Angabe sein.
endDate	DateTime	Ende des Zeitraums, für den der Trend der Verbindungsfehler abgefragt wird. Dies muss eine UTC-Angabe sein.
intervalLength	int	Intervall zwischen Datenpunkten in Minuten für den Trend.
desktopGroupsFilter	string	Durch Trennzeichen getrennte Liste der Bereitstellungsgruppen-UIDs, auf die das Ergebnis beschränkt wird.
connectionFailureTypeFilter	string	Durch Trennzeichen getrennte Liste der ganzen Zahlen für Verbindungsfehlertypen, auf die das Ergebnis beschränkt wird. Diese ganze Zahl entspricht der ConnectionFailureType-Enumeration.
sessionSupportFilter	string	Durch Trennzeichen getrennte Liste der Maschinentypen (Ganzzahl, Einzel = 1, Multi = 2, Unbekannt = 0), auf die das Ergebnis beschränkt wird. Erfolgt keine Angabe, werden beide Typen zurückgegeben.

Zurückgegebene Elemente

Ein IQueryable von TrendItem-Objekten.

Eigenschaft	Typ	Anmerkungen
FailureTrend	List	TrendItem enthält ein UTC-Datum (DateTime) und eine Anzahl (int) von Fehlern, die den Trend der Verbindungsfehler im angegebenen Zeitraum gemäß der festgelegten Filter darstellen.

Beispiele

Ruft einen 24-Stunden-Trend (Datenpunkt alle 15 Minuten) aller Typen von Fehlern bei Verbindungen mit Desktopbetriebssystemmaschinen für die beiden angegebenen Bereitstellungsgruppen ab.

`http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetConnectionFailureTrend?startdate=datetime'2011-09-30T00:00:00'&enddate=datetime'2011-10-`

Ruft einen Wochentrend (Datenpunkt alle 4 Stunden) für 3 Typen von Verbindungsfehlern (aggregiert) für alle Bereitstellungsgruppen ab, für die der Benutzer Zugriff hat.

`http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetConnectionFailureTrend?startdate=datetime'2011-09-23T00:00:00'&enddate=datetime'2011-10-01T00:00:00'intervalLength=240&connectionFailureTy`

Ruft eine Liste der Verbindungsfehler nach Typ sowie einen Trend für jeden Typ für den angegebenen Zeitraum ab. Jeder Datenpunkt im Trend repräsentiert die Gesamtzahl der Verbindungsfehler im letzten Intervall (intervalLength-Parameter). Die Abfrage kann auf einen bestimmten Fehlertyp beschränkt oder es können alle Fehler abgefragt werden.

Parametername	Typ	Anmerkungen
startDate	DateTime	Beginn des Zeitraums, für den der Trend der Verbindungsfehler abgefragt wird. Dies muss eine UTC-Angabe sein.
endDate	DateTime	Ende des Zeitraums, für den der Trend der Verbindungsfehler abgefragt wird. Dies muss eine UTC-Angabe sein.
intervalLength	Int	Intervall zwischen Datenpunkten in Minuten für den Trend.
connectionFailureType	Int	[ConnectionFailureType] Sofern festgelegt (>0), werden nur Verbindungsfehler dieses Typs zurückgegeben. Andernfalls (0) werden alle Fehlertypen zurückgegeben. Diese ganze Zahl entspricht der ConnectionFailureType-Enumeration.

Zurückgegebene Elemente

Ein IQueryable von ConnectionFailureTrend-Objekten, die für jeden ConnectionFailureType Folgendes enthalten.

Wenn alle Typen abgefragt werden, wird auch die Summe mit ConnectionFailureType = -1 zurückgegeben.

Enthält die Abfrage "Sexpand = FailureTrend", werden für jeden Fehlertyp die Eigenschaften "Trend" und "TotalFailureCount" abgerufen.

Enthält die Abfrage "Sexpand=DesktopGroupBreakdown", wird auch diese Eigenschaft abgerufen.

Eigenschaft	Typ	Anmerkungen
ConnectionFailureType	int [ConnectionFailureType]	Der Typ des Verbindungsfehlers (aus der Enumeration "ConnectionFailureType").
FailureTrend	List	TrendItem enthält ein UTC-Datum (DateTime) und einen Wert (nullable double), die den Trend der Verbindungsfehler des angegebenen Typs über den angegebenen Zeitraum darstellen.
TotalFailureCount	int	Die Gesamtzahl der Fehler im angegebenen Zeitraum.
DesktopGroupBreakdown	List	DesktopGroupBreakdown enthält ein DesktopGroup-Objekt (dieses muss im "expand" enthalten sein, damit es zurückgegeben wird) und Count (int), das die Gesamtzahl der Fehler im angegebenen Zeitraum darstellt.

Beispiele

Ruft unter Aufschlüsselung nach Bereitstellungsgruppe den Trend aller Fehlertypen für den Zeitraum vom 14. November 05:00 Uhr (UTC) bis zum 14. November 07:00 Uhr (UTC) mit einem Datenpunkt alle 30 Minuten ab:

`http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetConnectionFailureTrendsByType?startDate=datetime'2011-11-14T05:00:00'&endDate=datetime'2011-11-14T07:00:00'&intervalLength=30&connectionFailureType=4`
Ruft unter Aufschlüsselung nach Bereitstellungsgruppe den Trend des Fehlertyps 4 (keine Kapazität verfügbar) für den Zeitraum vom 14. November 05:00 Uhr (UTC) bis zum 14. November 07:00 Uhr (UTC) ab:

`http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetConnectionFailureTrendsByType?startDate=datetime'2011-11-14T05:00:00'&endDate=datetime'2011-11-14T07:00:00'&connectionFailureType=4&$expand=FailureTrend`
Ruft den Trend aller Fehlertypen für den Zeitraum vom 14. November 00:00 Uhr (UTC) bis zum 14. November 23:00 Uhr (UTC) mit einem Datenpunkt alle 60 Minuten ab:

`http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetConnectionFailureTrendsByType?startDate=datetime'2011-11-14T00:00:00'&endDate=datetime'2011-11-14T23:00:00'&intervalLength=60&connectionFailureType=0`

Aktualisiert: 2013-11-11

Ruft eine Liste der Verbindungsfehler nach Typ sowie einen Trend für jeden Typ für die angegebene Zahl Intervalle bis zum aktuellen Zeitpunkt (UTC-Serverzeit) ab. Jeder Datenpunkt im Trend repräsentiert die Gesamtzahl der Verbindungsfehler im letzten Intervall (intervalLength-Parameter). Die Abfrage kann auf einen bestimmten Fehlertyp beschränkt oder es können alle Fehler abgefragt werden.

Parametername	Typ	Anmerkungen
intervalLength	int	Intervall zwischen Datenpunkten in Minuten für den Trend.
numberOfIntervals	int	Anzahl der angeforderten Datenpunkte. Dient zusammen mit intervalLength zum Berechnen von StartDate.
connectionFailureType	int [ConnectionFailureType]	Sofern festgelegt (>0), werden nur Verbindungsfehler dieses Typs zurückgegeben. Andernfalls (0) werden alle Fehlertypen zurückgegeben. Diese ganze Zahl entspricht der ConnectionFailureType-Enumeration.

Zurückgegebene Elemente:

Ein IQueryable von ConnectionFailureTrend-Objekten, die für jeden ConnectionFailureType Folgendes enthalten.

Wenn alle Typen abgefragt werden, wird auch die Summe mit ConnectionFailureType = -1 zurückgegeben.

Enthält die Abfrage "Sexpand = FailureTrend", werden für jeden Fehlertyp die Eigenschaften "Trend" und "TotalFailureCount" abgerufen.

Enthält die Abfrage "Sexpand=DesktopGroupBreakdown", wird auch diese Eigenschaft abgerufen.

Eigenschaft	Typ	Anmerkungen
ConnectionFailureType	int [ConnectionFailureType]	Der Typ des Verbindungsfehlers (aus der Enumeration "ConnectionFailureType").
FailureTrend	List	Trenditem enthält ein UTC-Datum (DateTime) und einen Wert (nullable double), die den Trend der Verbindungsfehler des angegebenen Typs über den angegebenen Zeitraum darstellen.
TotalFailureCount	int	Die Gesamtzahl der Fehler im angegebenen Zeitraum.
DesktopGroupBreakdown	List	DesktopGroupBreakdown enthält ein DesktopGroup-Objekt (dieses muss im "expand" enthalten sein, damit es zurückgegeben wird) und Count (int), das die Gesamtzahl der Fehler im angegebenen Zeitraum darstellt.

Beispiele

Ruft unter Aufschlüsselung nach Bereitstellungsgruppe den Fehlertrend für die letzten 60 Minuten mit einem Datenpunkt pro Minute ab:

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetConnectionFailureTrendsByTypeLatest?&intervalLength=1&numberOfIntervals=60&connectionFailureType=0&\$expand=FailureTrend,DesktopGroupBreakdown

Ruft unter Aufschlüsselung nach Bereitstellungsgruppe den Trend des Fehlertyps 4 (keine Kapazität verfügbar) für die letzten 30 Minuten ab:

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetConnectionFailureTrendsByTypeLatest?&intervalLength=1&numberOfIntervals=30&connectionFailureType=4&\$expand=DesktopGroupBreakdown,DesktopGroupBreakdown

Ruft den Fehlertrend für alle Fehlertypen für die letzten 24 Stunden mit einem Datenpunkt pro Stunde ab:

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetConnectionFailureTrendsByTypeLatest?&intervalLength=60&numberOfIntervals=24&connectionFailureType=0&\$expand=FailureTrend

Ruft die Aufschlüsselung der Anmeldungen für Benutzer und Bereitstellungsgruppen ab. Gibt die aktuelle Benutzersitzung, sofern vorhanden, zurück.

Parametername	Typ	Verbindlich?	Anmerkungen
userSid	Zeichenfolge	Ja	ID des Benutzers, für den Informationen angefordert werden.
desktopGroupUid	GUID	Ja	Abgefragte Bereitstellungsgruppe.
machineUid	Zeichenfolge	Nein	Maschine, auf der die letzte Sitzung ausgeführt wurde. Dies ist möglicherweise nicht verfügbar.

Zurückgegebene Elemente

Ein IQueryable von LogOnBreakdown-Objekten, die für jeden Anmeldungsschritt Folgendes enthalten:

Eigenschaft	Typ	Anmerkungen
LogonStepItems	List	Liste der LogonStepItems. Jeder LogonStepItem besitzt einen LogonStep (mögliche Werte sind Brokering=1, VMStart=2, HDX=3, Authentication=4, Gpos=5, LogonScripts=6, ProfileLoad=7, Interactive=8, Total=0) und eine Duration (Nullable) in Millisekunden. Wenn keine Daten für einen Schritt verfügbar sind, wird die Dauer mit NULL zurückgegeben.
BreakdownType	int	Typ der Aufschlüsselung – UsersLastSession = 1, UsersSessionAverage = 2, DesktopGroupAverage = 3

Beispiel

Ruft die Anmeldeaufschlüsselung für Benutzer und Bereitstellungsgruppe ab. Gibt die aktuelle Benutzersitzung, falls vorhanden, zurück.

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetLastSessionLogOnBreakdown?userSid='User%20Sid'&desktopGroupUid=guid'28D15C1F-7D4C-43BC-BAF2-5886D8514642'&machineUid=guid'28D15C1F-7D4C-43BC-BAF2-5886D8514642'

Ruft den Load Evaluation Index-Trend (LEI) für Serverbetriebssystemmaschinen für den angegebenen Zeitraum (in Minuten) ab. Jeder Datenpunkt repräsentiert den durchschnittlichen Lastindex für das jeweilige Intervall. Es wird ein IQueryable von LoadIndexTrendItems zurückgegeben, das den effektiven Lastindex und die Aufschlüsselung nach einzelnen Typen (CPU, Arbeitsspeicher, Datenträger, Netzwerk, Sitzungszahl) darstellt.

Parametername	Typ	Anmerkungen
startDate	DateTime	Beginn des Zeitraums, für den der Lastindextrend abgefragt wird. Dies muss eine UTC-Angabe sein.
endDate	DateTime	Ende des Zeitraums, für den der Lastindextrend abgefragt wird. Dies muss eine UTC-Angabe sein.
intervalLength	int	Intervall zwischen Datenpunkten in Minuten für den Trend.
desktopGroupsFilter	string	Durch Trennzeichen getrennte Liste der Bereitstellungsgruppen-UIDs, auf die das Ergebnis beschränkt wird.

Parametername	Typ	Anmerkungen
machineFilter	string	Durch Trennzeichen getrennte Liste der Maschinen-SIDs, auf die das Ergebnis beschränkt wird. Erwartet MultiSession-Maschinen.

Zurückgegebene Elemente

Ein IQueryable von Objekten mit folgendem Inhalt:

Eigenschaft	Typ	Anmerkungen
LoadIndexTrend	List	LoadIndexTrendItem enthält ein UTC-Datum (DateTime) und eine ganze Zahl für den gesamten LEI sowie ganze Zahlen für die einzelnen LEI-Bestandteile (CPU, Arbeitsspeicher, Datenträger, Netzwerk, Sitzungszahl) in einem separaten Objekt (sofern "Sexpand=Breakdown" in der Abfrage festgelegt wurde).

Beispiele

Ruft einen 24-Stunden-Trend (Datenpunkt alle 15 Minuten) des Lastindexrends unter Aufschlüsselung nach den beiden angegebenen Bereitstellungsgruppen ab:

http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetEffectiveLoadIndexTrend?startDate=datetime'2011-09-30T00:00:00'&endDate=datetime'2011-10-01T00:00:00'intervalLength=15&desktopGroupsFilter=Ruft einen wöchentlichen Lastindextrend (Datenpunkt alle 4 Stunden) ab:

http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetEffectiveLoadIndexTrend?startDate=datetime'2011-09-23T00:00:00'&endDate=datetime'2011-10-01T00:00:00'intervalLength=240&\$expand=Breakdo

Aktualisiert: 2013-07-24

Ruft den Load Evaluation Index-Trend (LEI) für Serverbetriebssystemmaschinen für den angegebenen Zeitraum nach LEI-Typen ab. Jeder Datenpunkt im Trend zeigt den gesamten LEI sowie die Aufschlüsselung nach Lastindextyp (CPU, Arbeitsspeicher, Datenträger, Netzwerk, Sitzungszahl).

Parametername	Typ	Anmerkungen
startDate	DateTime	Beginn des Zeitraums, für den der Lastindextrend abgefragt wird. Dies muss eine UTC-Angabe sein.
endDate	DateTime	Ende des Zeitraums, für den der Lastindextrend abgefragt wird. Dies muss eine UTC-Angabe sein.
intervalLength	int	Intervall zwischen Datenpunkten in Minuten für den Trend
desktopGroupsFilter	string	Durch Trennzeichen getrennte Liste der Bereitstellungsgruppen-UIDs, auf die das Ergebnis beschränkt wird.
machineFilter	string	Durch Trennzeichen getrennte Liste der Maschinen-SIDs, auf die das Ergebnis beschränkt wird. Erwartet MultiSession-Maschinen.
indexTypeFilter	string	Durch Trennzeichen getrennte Liste der einzuschließenden Lastindextypen (effektiv, CPU, Netzwerk, Arbeitsspeicher, Datenträger Sitzungszahl). Entspricht der LoadIndexType-Enumeration. Eine leere oder NULL-Zeichenfolge gibt alle Typen zurück.

Zurückgegebene Elemente

Ein IQueryable von Objekten, das für jeden festgelegten Indextyp Folgendes enthält:

Eigenschaft	Typ	Anmerkungen
LoadIndexType	int	[LoadIndexType] entspricht der LoadIndexType-Enumeration.
LoadIndexTrend	List	TrendItem enthält ein UTC-Datum (DateTime) und eine ganze Zahl für den Index.

Beispiele

Ruft einen 24-Stunden-Trend (Datenpunkt alle 15 Minuten) des Lastindexrends unter Aufschlüsselung nach den beiden angegebenen Bereitstellungsgruppen ab:

http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetLoadIndexTrend?startDate=datetime'2011-09-30T00:00:00'&endDate=datetime'2011-10-01T00:00:00'intervalLength=15&desktopGroupsFilter='04D5'Ruft einen LEI-Wochentrend (Datenpunkt alle 4 Stunden) unter Aufschlüsselung ab:

http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetLoadIndexTrend?startDate=datetime'2011-09-23T00:00:00'&endDate=datetime'2011-10-01T00:00:00'intervalLength=240

Ruft den Anmeldedauertrend (Durchschnitt) gruppiert nach Intervall (Uhrzeit, Wochentag, etc.) für den angegebenen Zeitraum ab. Abgedeckt werden nur erfolgreiche Anmeldungen (d. h. die einen LogOnStartDate- und einen LogOnEndDate-Eintrag in der Datenbank haben).

Parametername	Typ	Anmerkungen
start	DateTime	Beginn des Zeitraums, für den der durchschnittliche Anmeldedauertrend abgefragt wird. Dies muss eine UTC-Angabe sein.
end	DateTime	Ende des Zeitraums, für den der Anmeldedauertrend abgefragt wird. Dies muss eine UTC-Angabe sein.
intervalLength	int	Intervall zwischen Datenpunkten in Minuten für den Trend.

Zurückgegebene Elemente

Ein IQueryable von TrendItem-Objekten. TrendItem enthält ein UTC-Datum(DateTime) und einen Wert zur Darstellung des Trends durchschnittlicher Anmeldezahlen in dem festgelegten Zeitraum.

Beispiel

Ruft Daten für einen Zeitraum von einer Stunde ab mit einem Datenpunkt pro Minute, der die Gesamtzahl der Anmeldungen für diese Minute darstellt. Gab es während der betreffenden Minute keine Anmeldungen, ist der Wert im TrendItem "Null".

```
http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetLogOnCountTrend?startDate=datetime'2012-06-05T12:24:00'&endDate=datetime'2012-06-05T13:24:00'&intervalLength=1
```

Ruft den Trend der Anmeldezahlen für den angegebenen Zeitraum ab. Jeder Datenpunkt im Trend repräsentiert die Gesamtzahl der Anmeldungen im letzten Intervall (intervalLength-Parameter). Abgedeckt werden nur erfolgreiche Anmeldungen (d. h. die einen LogOnStartDate- und einen LogOnEndDate-Eintrag in der Datenbank haben).

Parametername	Typ	Anmerkungen
startDate	DateTime	Beginn des Zeitraums, für den der Trend der Anmeldeanzahl abgefragt wird. Dies muss eine UTC-Angabe sein.
endDate	DateTime	Ende des Zeitraums, für den der Trend der Anmeldeanzahl abgefragt wird. Dies muss eine UTC-Angabe sein.
intervalLength	int	Intervall zwischen Datenpunkten in Minuten für den Trend.
desktopGroupsFilter	string	Durch Trennzeichen getrennte Liste der Desktopgruppen-UIDs, auf die das Ergebnis beschränkt wird.

Zurückgegebene Elemente

Eine Liste mit TrendItems:

Eigenschaft	Typ	Anmerkungen
LogonCountTrend	List	Das LogonDurationTrendItem enthält ein UTC-Datum(DateTime) und eine Zahl (int) für die Anzahl der Anmeldungen zu dem festgelegten Datum.

Beispiel

Ruft einen 24-Stunden-Trend (Datenpunkt alle 15 Minuten) der Anmeldeanzahlen (in Millisekunden) für die beiden festgelegten Bereitstellungsgruppen ab:

```
http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetLogOnCountTrendFiltered?start=datetime'2011-09-30T00:00:00'&end=datetime'2011-10-01T00:00:00'&intervalLength=15&desktopGroupsFilter='04D5:  
Ruft einen Wochentrend (Datenpunkt alle 4 Stunden) des Trends der Anmeldedauer (in Millisekunden) ab:
```

```
http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetLogOnCountTrendFiltered?start=datetime'2011-09-23T00:00:00'&end=datetime'2011-10-01T00:00:00'&intervalLength=240
```

Ruft den Trend der durchschnittlichen Anmeldedauer für den angegebenen Zeitraum ab, wobei die Durchschnittswerte nach Intervall gruppiert werden. Jeder Datenpunkt liefert die durchschnittliche Anmeldedauer und eine Aufschlüsselung nach Intervall gemäß groupByInterval.

Parametername	Typ	Anmerkungen
start	DateTime	Beginn des Zeitraums, für den der durchschnittliche Anmeldedauertrend abgefragt wird. Dies muss eine UTC-Angabe sein.
end	DateTime	Ende des Zeitraums, für den der Anmeldedauertrend abgefragt wird. Dies muss eine UTC-Angabe sein.
groupByInterval	int	Art des für die Aufschlüsselung verwendeten Intervalls. Optionen: TimeOfDay=1, DayOfWeek=2, Month=3, Year=4. Pflichteingabe.
desktopGroupsFilter	string	Durch Trennzeichen getrennte Liste der Bereitstellungsgruppen-UIDs, auf die das Ergebnis beschränkt wird.

Zurückgegebene Elemente

Eine Liste von GroupedLogOnDurationTrendItem

Eigenschaft	Typ	Anmerkungen
GroupByDate	int	Repräsentiert die Aufschlüsselung (4 = 04:00 Uhr, 2007 = Jahr 2007, abhängig von der Eingabe für groupByInterval). Zahl zurückgegebener Datenpunkte: GroupByInterval=TimeOfDay (1): 24 Punkte (0-23) 0 = 00:00 Uhr, 1 = 01:00 Uhr, 2 = 02:00 Uhr... 23 = 23:00 Uhr GroupByInterval=DayOfWeek (2): 7 Punkte (0-6) 0 = Sonntag, 1 = Montag... 6 = Samstag GroupByInterval=Month (3): 12 Punkte (0-11) 0 = Januar, 1 = Februar... 11 = Dezember GroupByInterval=Year (4): je nach Anzahl der Jahre innerhalb des startDate/endDate-Bereichs.

Eigenschaft	Typ	Anmerkungen
Value Nullable		Durchschnittliche Gesamt-Anmeldedauer in Millisekunden und durchschnittliche Dauer der einzelnen Schritte für die Gruppe.
Breakdown	Collection	Sammlung von LogOnStepItem-Objekten, die Folgendes enthalten: Duration (Nullable), d. h. die durchschnittliche Dauer des Schritts in Millisekunden, und LogOnStep (int), d. h. der ganzzahlige Enumerationswert des Anmeldeschritts (bezieht sich auf die LogOnStep-Enumeration).

Beispiel

Ruft einen 7-Tage-Durchschnitt nach Tageszeit (Stunde) des Anmeldedauertrends (in Millisekunden) für die beiden festgelegten Bereitstellungsgruppen ab. Damit wird ein Trenditem für jede Stunde des Tages mit der durchschnittlichen Anmeldedauer für diese Stunde für den festgelegten Zeitraum (24 Datenpunkte) zurückgegeben.

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetDurationAverageTrend?start=datetime'2011-09-23T00:00:00'&end=datetime'2011-10-01T00:00:00'groupByInterval=1&desktopGroupsFilter='04D53Bf

Ruft den Trend der Anmeldedauer für den angegebenen Zeitraum ab. Jeder Datenpunkt liefert die Gesamtdauer der Anmeldung und eine Aufschlüsselung für jeden Schritt.

Parametername	Typ	Anmerkungen
startDate	DateTime	Beginn des Zeitraums, für den der Trend der Verbindungsfehler abgefragt wird. Dies muss eine UTC-Angabe sein.
endDate	DateTime	Ende des Zeitraums, für den der Trend der Verbindungsfehler abgefragt wird. Dies muss eine UTC-Angabe sein.
intervalLength	int	Intervall zwischen Datenpunkten in Minuten für den Trend.
desktopGroupsFilter	string	Durch Trennzeichen getrennte Liste der Bereitstellungsgruppen-UIDs, auf die das Ergebnis beschränkt wird.

Zurückgegebene Elemente

Eine Liste mit LogonDurationTrendItems:

Eigenschaft	Typ	Anmerkungen
LogonDurationTrend	List	LogOnDurationTrendItem enthält ein UTC-Datum (DateTime) und ein Double für die durchschnittliche Anmeldedauer in Millisekunden während des Zeitraums. Außerdem enthält es eine Aufschlüsselung der Dauer der einzelnen Anmeldeschritte für jeden Datenpunkt.

Beispiel

Ruft einen 24-Stunden-Trend (Datenpunkt alle 15 Minuten) der Anmeldedauer (in Millisekunden) für die beiden festgelegten Bereitstellungsgruppen ab:

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetLogOnDurationDetailsTrend?startDate=datetime'2011-09-30T00:00:00'&endDate=datetime'2011-10-01T00:00:00'intervalLength=15&desktopGroupsF

Ruft einen Wochentrend (Datenpunkt alle 4 Stunden) des Trends der Anmeldedauer (in Millisekunden) ab:

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetLogOnDurationDetailsTrend?startDate=datetime'2011-09-23T00:00:00'&endDate=datetime'2011-10-01T00:00:00'intervalLength=240

Ruft den Trend der Anmeldedauer (Durchschnitt) für den angegebenen Zeitraum ab.

Parametername	Typ	Anmerkungen
start	DateTime	Beginn des Zeitraums, für den der durchschnittliche Anmeldedauertrend abgefragt wird. Dies muss eine UTC-Angabe sein.
end	DateTime	Ende des Zeitraums, für den der durchschnittliche Anmeldedauertrend abgefragt wird. Dies muss eine UTC-Angabe sein.
intervalLength	int	Intervall zwischen Datenpunkten in Minuten für den Trend.

Zurückgegebene Elemente

Ein IQueryable von TrendItem-Objekten. TrendItem enthält ein UTC-Datum (DateTime) und einen Wert (in Millisekunden), der den Trend der durchschnittlichen Anmeldedauer im angegebenen Zeitraum darstellt.

Beispiele

Ruft Daten für einen Zeitraum von einer Stunde ab mit einem Datenpunkt pro Minute, der die durchschnittliche Anmeldedauer für die Minute darstellt. Gab es während der betreffenden Minute keine Anmeldungen , ist der Wert im Trenditem "NaN" (keine Zahl).

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetLogOnDurationTrend?start=datetime'2012-06-05T12:24:00'&end=datetime'2012-06-05T13:24:00'&intervalLength=1

Ruft einen einzelnen Datenpunkt ab, der die durchschnittliche Anmeldedauer für die Stunde darstellt.

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetLogOnDurationTrend?start=datetime'2012-06-05T12:24:00'&end=datetime'2012-06-05T13:23:00'&intervalLength=60

Ruft den Trend der Anzahl gleichzeitiger Maschinenfehler für den angegebenen Zeitraum ab. Jeder Datenpunkt repräsentiert die maximale Anzahl der Maschinen mit einem Fehler zu einem beliebigen

Zeitpunkt im letzten Intervall (intervalLength-Parameter). Die Abfrage kann auf einen bestimmten Maschinenfehlertyp beschränkt oder es können alle Fehler abgefragt werden.

Parametername	Typ	Anmerkungen
startDate	DateTime	Beginn des Zeitraums, für den der Trend der Verbindungsfehler abgefragt wird. Dies muss eine UTC-Angabe sein.
endDate	DateTime	Ende des Zeitraums, für den der Trend der Verbindungsfehler abgefragt wird. Dies muss eine UTC-Angabe sein.
intervalLength	int	Intervall zwischen Datenpunkten in Minuten für den Trend.
desktopGroupsFilter	string	Durch Trennzeichen getrennte Liste der Bereitstellungsgruppen-UIDs, auf die das Ergebnis beschränkt wird.
machineFailureTypeFilter	string	Durch Trennzeichen getrennte Liste der ganzen Zahlen für Verbindungsfehlertypen, auf die das Ergebnis beschränkt wird. Diese ganze Zahl entspricht der MachineFaultStateCode-Enumeration. Wird dies nicht angegeben, werden alle Typen im Trend aggregiert zurückgegeben.
sessionSupportFilter	int	Durch Trennzeichen getrennte Liste der Maschinentypen (ganze Zahl, Einzel = 1, Multi = 2, Unbekannt = 0), auf die das Ergebnis beschränkt wird. Erfolgt keine Angabe, werden alle Typen zurückgegeben.

Zurückgegebene Elemente

Eine Liste von TrendItem-Objekten (FailureTrend).

Eigenschaft	Typ	Anmerkungen
FailureTrend	List	TrendItem enthält ein UTC-Datum (DateTime) und eine Anzahl (int) von Fehlern, die den Trend der Maschinenfehler im angegebenen Zeitraum gemäß der festgelegten Filter darstellen.

Beispiel

Ruft einen 24-Stunden-Trend (Datenpunkt alle 15 Minuten) aller Typen von Maschinenfehlern mit Desktopbetriebssystemmaschinen für die beiden angegebenen Bereitstellungsgruppen ab.

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetMachineFailureTrend?startDate=datetime'2011-09-30T00:00:00'&endDate=datetime'2011-10-01T00:00:00'&intervalLength=15&sessionSupportFilter=""
Ruft einen 24-Stunden-Trend (Datenpunkt alle 2 Stunden) der Maschinenfehler des Typs "Konnte nicht gestartet werden" und "Beim Starten hängen geblieben" für alle Bereitstellungsgruppen und alle Sitzungstypen ab.

http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetMachineFailureTrend?startDate=datetime'2011-09-23T00:00:00'&endDate=datetime'2011-10-01T00:00:00'&intervalLength=240&machineFailureTypeF

Ruft eine Liste der Maschinenfehler- und Erfolgsquoten nach Typ mit einem Trend für jeden Typ für den angegebenen Zeitraum ab. Jeder Datenpunkt repräsentiert die Gesamtanzahl der Maschinen in einem Fehlerstatus im letzten Intervall (intervalLength-Parameter). Die Abfrage kann auf einen bestimmten Fehlertyp beschränkt oder es können alle Fehler abgefragt werden.

Parametername	Typ	Anmerkungen
startDate	DateTime	Beginn des Zeitraums, für den der Trend der Maschinenfehler abgefragt wird. Dies muss eine UTC-Angabe sein. Nicht erforderlich, wenn keine Trends abgerufen werden.
endDate	DateTime	Ende des Zeitraums, für den der Trend der Maschinenfehler abgefragt wird. Dies muss eine UTC-Angabe sein.
intervalLength	int	Intervall zwischen Datenpunkten in Minuten für den Trend.
machineFailureType	int (MachineFailureType)	Sofern festgelegt (>0), werden nur Maschinenfehler dieses Typs zurückgegeben. Andernfalls (0) werden alle Fehlertypen zurückgegeben. Diese ganze Zahl entspricht der MachineFailureType-Enumeration.
sessionSupport	int (SessionSupport)	Nullable int. Sofern nicht angegeben, werden alle sessionSupport-Typen zurückgegeben. Einzel = 1, Multi = 2, Unbekannt = 0

Zurückgegebene Elemente

Ein IQueryable von MachineFailureTrend-Objekten, die für jeden MachineFailureTrend Folgendes enthalten.

Wenn alle Typen abgefragt werden, wird auch die Summe mit MachineFailureTrend = -1 zurückgegeben.

Enthält die Abfrage "Sexpand = FailureTrend", werden für jeden Fehlertyp die Eigenschaften "Trend" und "TotalFailureCount" abgerufen.

Enthält die Abfrage "Sexpand=DesktopGroupBreakdown", wird auch diese Eigenschaft abgerufen.

Eigenschaft	Typ	Anmerkungen
MachineFailureType	int [MachineFailureType]	Der Typ des Verbindungsfehlers (siehe Bestimmen der Enumerationswerte).
TotalFailureCount	int	Die Gesamtzahl der Fehler zum endDate-Datum.
FailureTrend	List	TrendItem enthält ein UTC-Datum (DateTime) und ein Count (double), das den Trend der Maschinenfehler des angegebenen Typs über den angegebenen Zeitraum darstellen.
DesktopGroupBreakdown	List	DesktopGroupBreakdown enthält ein DesktopGroup-Objekt und ein Count (int), das die Gesamtzahl der Maschinen in einem Fehlerzustand

Eigenschaft	Typ	Anmerkungen
-------------	-----	-------------

Beispiele

Ruft unter Aufschlüsselung nach Bereitstellungsgruppe den Trend aller Maschinenfehlertypen für den Zeitraum vom 14. November 05:00 Uhr (UTC) bis zum 14. November 07:00 Uhr (UTC) mit einem Datenpunkt alle 30 Minuten für alle Serverbetriebssystemmaschinen (d. h. sessionSupport = 2) ab:

`http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetMachineFailureTrendsByType?startDate=datetime'2011-11-14T05:00:00'&endDate=datetime'2011-11-14T07:00:00'&intervalLength=30&machineFailureType=3`
 Ruft unter Aufschlüsselung nach Bereitstellungsgruppe den Trend des Maschinenfehlertyps 3 (beim Starten hängen geblieben) für den Zeitraum vom 14. November 05:00 Uhr (UTC) bis zum 14. November 07:00 Uhr (UTC) für Desktopbetriebssystemmaschinen ab:

`http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetMachineFailureTrendsByType?startDate=datetime'2011-11-14T05:00:00'&endDate=datetime'2011-11-14T07:00:00'&machineFailureType=3&sessionSupport=2`
 Ruft den Trend aller Fehlertypen für alle Maschinensitzungen für den Zeitraum vom 14. November 00:00 Uhr (UTC) bis zum 14. November 23:00 Uhr (UTC) mit einem Datenpunkt alle 60 Minuten ab:

`http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetMachineFailureTrendsByType?startDate=datetime'2011-11-14T00:00:00'&endDate=datetime'2011-11-14T23:00:00'&intervalLength=60&machineFailureType=3`

Ruft Datenpunkte für den angegebenen Zeitraum ab zur Darstellung eines Trends der gleichzeitigen (verbundenen) Sitzungen in Form einer Liste von DateTime/Wert-Paaren. Jeder Punkt stellt die maximale Anzahl gleichzeitig verbundener Sitzungen während des angegebenen Intervalls dar. Dieser Wert wird als Summe von DesktopGroups und MAX der Summe berechnet.

Parametername	Typ	Anmerkungen
startDate	DateTime	Beginn des Zeitraums, für den die gleichzeitigen Sitzungen abgefragt werden. Dies muss eine UTC-Angabe sein.
endDate	DateTime	Ende des Zeitraums, für den die gleichzeitigen Sitzungen abgefragt werden. Dies muss eine UTC-Angabe sein.
intervalLength	int	Intervall zwischen Datenpunkten in Minuten für den Trend.
sessionSupportFilter	string	Durch Trennzeichen getrennte Liste der Maschinentypen (ganze Zahl, Einzel = 1, Multi = 2, Unbekannt = 0), auf die die Abfrage beschränkt wird. Erfolgt keine Angabe, werden alle Typen zurückgegeben.
connectionStateFilter	string	Liste durch Trennzeichen getrennter Verbindungsstatus (ganze Zahl, (1 = verbunden, 2 = getrennt), auf die die Abfrage beschränkt wird. Erfolgt keine Angabe, werden beide zurückgegeben.
desktopGroupFilter	string	Durch Trennzeichen getrennte Liste der Bereitstellungsgruppen-UIDs, auf die die Abfrage beschränkt wird. Erfolgt keine Angabe, werden alle Bereitstellungsgruppen zurückgegeben.

Zurückgegebene Elemente

IQueryable-Liste von TrendItem-Objekten – IQueryable, aufsteigend nach Datum sortiert.

Jedes TrendItem-Objekt enthält eine UTC-Datumeigenschaft (DateTime) und eine Werteigenschaft (int) zum Zeichnen des Trends.

Beispiel

Ruft einen 24-Stunden-Trend mit Datenpunkten für jede Stunde für gleichzeitig verbundene Sitzungen ab. Jeder Datenpunkt repräsentiert die maximale Anzahl von gleichzeitigen (nur verbundenen) Sitzungen im letzten Intervall (in diesem Fall 60 Minuten).

`http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetSessionCountTrend?startDate=datetime'2011-09-30T00:00:00'&endDate=datetime'2011-10-01T00:00:00'&intervalLength=60&connectionStateFilter=1`
 Ruft einen 24-Stunden-Trend mit Datenpunkten für jede Stunde für gleichzeitig getrennte Sitzungen ab. Jeder Datenpunkt repräsentiert die maximale Anzahl von gleichzeitigen (nur getrennten) Sitzungen im letzten Intervall (in diesem Fall 60 Minuten).

`http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetSessionCountTrend?startDate=datetime'2011-09-30T00:00:00'&endDate=datetime'2011-10-01T00:00:00'&intervalLength=60&connectionStateFilter=2`
 Ruft den Trend der vergangenen Woche ab, mit Datenpunkten alle 6 Stunden für gleichzeitige Sitzungen (verbunden und getrennt) für beide Maschinentypen (Desktopbetriebssystem- und Serverbetriebssystemmaschinen). Jeder Datenpunkt repräsentiert die maximale Anzahl von gleichzeitigen Sitzungen im letzten Intervall (in diesem Fall 6 Stunden).

`http://{dc-host}/Citrix/Monitor/OData/v1/Methods/GetSessionCountTrend?startDate=datetime'2011-09-23T00:00:00'&endDate=datetime'2011-10-01T00:00:00'&intervalLength=360`

Ruft Datenpunkte für den angegebenen Zeitraum ab zur Darstellung eines Trends der gleichzeitigen (verbundenen) Sitzungen in Form einer Liste von DateTime/Wert-Paaren. Jeder Punkt stellt die maximale Anzahl gleichzeitig verbundener Sitzungen während des angegebenen Intervalls dar.

Parametername	Typ	Anmerkungen
startDate	DateTime	Beginn des Zeitraums, für den die gleichzeitigen Sitzungen abgefragt werden. Dies muss eine UTC-Angabe sein.
endDate	DateTime	Ende des Zeitraums, für den die gleichzeitigen Sitzungen abgefragt werden. Dies muss eine UTC-Angabe sein.
sessionSupportFilter	string	Durch Trennzeichen getrennte Liste der Maschinentypen (ganze Zahl, Einzel = 1, Multi = 2, Unbekannt = 0), auf die die Abfrage beschränkt wird. Standardmäßig werden alle Typen zurückgegeben, wenn kein Filter festgelegt wird.
desktopGroupFilter	string	Durch Trennzeichen getrennte Liste der Bereitstellungsgruppen-UIDs, auf die die Abfrage beschränkt wird. Wenn kein Wert angegeben wird, werden alle Bereitstellungsgruppen zurückgegeben.
connectionStateFilter	string	Liste durch Trennzeichen getrennter Verbindungsstatus (ganze Zahl, (1 = verbunden, 2 = getrennt), auf die die Abfrage beschränkt wird. Erfolgt keine Angabe, werden beide zurückgegeben.

Zurückgegebene Elemente

IQueryable-Liste von SessionSummary-Objekten – IQueryable.

Jedes SessionSummary-Objekt enthält Folgendes:

- DesktopGroup (name und Uid)
- AverageSessionCount (double)
- Durchschnittliche Anzahl gleichzeitiger Sitzungen im angegebenen Zeitraum. Dies ist der Durchschnitt der Spitzenwerte in dem Zeitraum mit der für den Zeitraum möglichen Granularität.
- AverageUserCount (double)
- Durchschnittliche Anzahl der verbundenen Benutzer im angegebenen Zeitraum
- PeakUserCount (int)
- Spitzenwert der Anzahl der zu einem beliebigen Zeitpunkt im angegebenen Zeitraum verbundenen Benutzer
- UniqueUserCount (int)
- Die Gesamtzahl der eindeutigen Benutzer im angegebenen Zeitraum
- AverageHoursPerDay (double)
- Durchschnittliche Sitzungsdauer (Stunden pro Benutzer pro Tag)
- DesktopGroup (name und Uid)

Beispiel

Ruft eine Liste der SessionSummary-Objekte innerhalb von 24 Stunden für alle Sitzungen ab.

`http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetSessionSummary?startDate=datetime'2011-09-30T00:00:00'&endDate=datetime'2011-10-01T00:00:00'`

Ruft eine Liste der SessionSummary-Objekte innerhalb von 24 Stunden für Serverbetriebssystemmaschinen ab.

`http://(dc-host)/Citrix/Monitor/OData/v1/Methods/GetSessionSummary?startDate=datetime'2011-09-30T00:00:00'&endDate=datetime'2011-10-01T00:00:00'&sessionSupportFilter='2'`

Bestimmen der Enumerationswerte

Sep 29, 2015

In diesem Abschnitt wird erläutert, wie die Enumerationstypen und -werte, die von der Überwachungsdienst-OData-API zurückgegeben werden, bestimmt werden.

Einige der Methoden der Überwachungsdienst-OData-API geben Enumerationen, wie z. B. `GetSessionCountTrend` und `GetMachineFailureTrend`, zurück. Zum Bestimmen der Werte dieser Enumerationen verwenden Sie die folgenden "Metadaten"-OData-APIs:

- **GetAllMonitoringEnums** gibt `IQueryable` zurück, wobei `MonitoringEnumeration` eine Klasse ist, die einen Enumerationstypnamen und einen Wertesatz umfasst.
`public IQueryable GetAllMonitoringEnums()`
- **GetMonitoringEnum** gibt den Wertesatz für den spezifischen angegebenen Typ zurück.
`public IQueryable GetMonitoringEnum(string typeName)`

Die folgenden Enumerationen werden als Parameter in den Überwachungsdienst-OData-APIs verwendet:

sessionSupportFilter

- Zuordnung zur Enumeration "SessionSupportCode"
- verwendet in:
 - `GetSessionCountTrend`
 - `GetSessionSummary`
 - `GetConnectedUsersTrend`
 - `GetConnectionFailureTrend`
 - `GetMachineFailureTrend`
 - `GetMachineFailureTrendsByType`
- Werte:
 - 0 = Unbekannt (Platzhalter - nicht verwenden)
 - 1 = Einzelsitzung (Desktopbetriebssystemmaschinen)
 - 2 = Multisitzung (Serverbetriebssystemmaschinen)

connectionStateFilter

- Zuordnung zur Enumeration "ConnectionState"
- verwendet in:
 - `GetSessionCountTrend`
 - `GetSessionSummary`
- Werte:
 - 0 = Unbekannt (Platzhalter - nicht verwenden)
 - 1 = Verbunden
 - 2 = Getrennt
 - 3 = Beendet
 - 4 = Wird vorbereitet
 - 5 = Aktiv
 - 6 = Verbindung wiederherstellen
 - 7 = Sitzung ohne Broker

- 8 = Andere
- 9 = Ausstehend

connectionFailureFilter

- Zuordnung zur Enumeration "ConnectionFailureType"
- verwendet in:
 - GetConnectionFailureTrendsByType
 - GetConnectionFailureTrendsByTypeLatest
 - GetConnectionFailureTrend
- Werte:
 - 0 = Keine
 - 1 = Clientverbindungsfehler
 - 2 = Maschinenfehler
 - 3 = Keine Kapazität verfügbar
 - 4 = Keine Lizenz verfügbar
 - 5 = Konfiguration

machineFailureTypeFilter

- Zuordnung zur Enumeration "MachineFaultStateCode"
- verwendet in:
 - GetMachineFailureTrend
 - GetMachineFailureTrendsByType
 - GetMachineFailureTrendsByTypeLatest
- Werte:
 - 0 = Unbekannt (Platzhalter - nicht verwenden)
 - 1 = Keine
 - 2 = Start fehlgeschlagen
 - 3 = Bei Start hängengeblieben
 - 4 = Nicht registriert
 - 5 = Maximale Kapazität

Beispiele

Sep 29, 2015

Die folgenden Beispiele zeigen, wie Daten mit der Überwachungsdienst-OData-API exportiert werden. Dieser Abschnitt enthält auch eine Liste von URLs für die verfügbaren Datengruppen.

Beispiel 1 – unformatiertes XML

1. Geben Sie die URL für jede Datengruppe in einen Webbrowser ein, der mit den für die XenApp- bzw. XenDesktop-Site erforderlichen Administratorrechten ausgeführt wird. Citrix empfiehlt die Verwendung des Chrome-Browsers mit dem Add-In Advanced Rest Client.
2. Zeigen Sie die Quelle an.

Beispiel 2 - PowerPivot für Excel

1. Installieren Sie Microsoft Excel.
2. Folgen Sie den hier verfügbaren Anweisungen zum Installieren von PowerPivot (abhängig davon, ob Sie Version 2010 oder 2013 verwenden): <https://support.office.com/en-us/article/Start-Power-Pivot-in-Microsoft-Excel-2013-add-in-a891a66d-36e3-43fc-81e8-fc4798f39ea8>.
3. Öffnen Sie Excel mit den für die XenApp- bzw. XenDesktop-Site erforderlichen Administratorrechten.

• Bei Verwendung von Excel 2010

1. Klicken Sie auf der Registerkarte "PowerPivot".
2. Klicken Sie auf "PowerPivot-Fenster".
3. Klicken Sie im Menüband auf **Aus Datenfeeds**.
4. Wählen Sie einen Anzeigenamen für die Verbindung aus (z. B. "XenDesktop-Überwachungsdaten") und geben Sie die Datenfeed-URL ein: `http://{dc-host}/Citrix/Monitor/OData/v1/Data` (oder "https:", falls Sie SSL verwenden).
5. Klicken Sie auf **Weiter**.
6. Wählen Sie die Tabellen aus, die Sie in Excel importieren möchten, und klicken Sie auf **Fertig stellen**. Die Daten werden abgerufen.

• Bei Verwendung von Excel 2013

1. Klicken Sie auf die Registerkarte "Daten".
2. Wählen Sie "Aus anderen Quellen > Aus OData-Datenfeed".
3. Geben Sie die URL des Datenfeeds ein: `http://{dc-host}/Citrix/Monitor/OData/v1/Data` (oder "https:", wenn Sie SSL verwenden) und klicken Sie auf **Weiter**.
4. Wählen Sie die Tabellen aus, die Sie in Excel importieren möchten, und klicken Sie auf **Weiter**.
5. Akzeptieren Sie die Standardnamen oder passen Sie Namen an und klicken Sie auf **Fertig stellen**.
6. Wählen Sie **Nur Verbindung** oder **Pivot Report**. Die Daten werden abgerufen.

Sie können die Daten jetzt über PowerPivot mit PivotTables und PivotCharts anzeigen und analysieren. Weitere Informationen finden Sie im Learning Center: <http://www.microsoft.com/en-us/bi/LearningCenter.aspx>.

Beispiel 3 – LinqPad

1. Laden Sie die neueste Version von LINQPad von <http://www.linqpad.net> herunter und installieren Sie sie.
2. Führen Sie LinqPad mit den für die XenApp- bzw. XenDesktop-Site erforderlichen Administratorrechten aus.

Tipp: Die einfachste Methode ist Download, Installation und Ausführung auf dem Delivery Controller.

3. Klicken Sie auf den Link "Add connection".
4. Wählen Sie "WCF Data Services 5.1 (OData 3)" und klicken Sie auf **Weiter**.
5. Geben Sie die URL des Datenfeeds ein: `http://{dc-host}/Citrix/Monitor/OData/v1/Data` (oder "https:", wenn Sie SSL verwenden). Falls erforderlich, geben Sie den Benutzernamen und das Kennwort für den Zugriff auf den Delivery Controller ein. Klicken Sie auf **OK**.
6. Sie können jetzt LINQ-Abfragen für den Datenfeed ausführen und die Daten nach Bedarf exportieren. Beispiel: Klicken Sie mit der rechten Maustaste auf "Catalogs" und wählen Sie **Catalogs.Take(100)** aus. Damit werden die ersten 100 Kataloge in der Datenbank zurückgegeben. Wählen Sie "Export > Export to Excel with formatting" aus.

URLs für verfügbare Datengruppen

URL	Beschreibung
<code>http://{dc-host}/Citrix/Monitor/OData/v1/Data/Catalogs</code>	Katalogimages in der Site
<code>http://{dc-host}/Citrix/Monitor/OData/v1/Data/ConnectionFailureCategories</code>	Kategorien für Verbindungsfehlertypen
<code>http://{dc-host}/Citrix/Monitor/OData/v1/Data/ConnectionFailureLogs</code>	Protokoll der einzelnen Verbindungsfehler in der Site
<code>http://{dc-host}/Citrix/Monitor/OData/v1/Data/Connections</code>	Repräsentiert eine erste Verbindung oder eine erneute Verbindung für eine Sitzung
<code>http://{dc-host}/Citrix/Monitor/OData/v1/Data/DesktopGroups</code>	Bereitstellungsgruppen in der Site
<code>http://{dc-host}/Citrix/Monitor/OData/v1/Data/FailureLogSummaries</code>	Fehlerzahlen (Verbindung/Maschine) nach Zeitraum und Bereitstellungsgruppe
<code>http://{dc-host}/Citrix/Monitor/OData/v1/Data/Hypervisors</code>	Hosts (Hypervisors) in der Site
<code>http://{dc-host}/Citrix/Monitor/OData/v1/Data/LoadIndexes</code>	Lastindexdaten, die vom Virtual Delivery Agent (VDA) empfangen wurden
<code>http://{dc-host}/Citrix/Monitor/OData/v1/Data/LoadIndexSummaries</code>	Lastindexdurchschnitte nach Zeitraum und Maschine
<code>http://{dc-host}/Citrix/Monitor/OData/v1/Data/MachineFailureLogs</code>	Protokoll jedes Maschinenfehlers in der Site nach Start- und Enddatum
<code>http://{dc-host}/Citrix/Monitor/OData/v1/Data/Machines</code>	Maschinen in der Site
<code>http://{dc-host}/Citrix/Monitor/OData/v1/Data/SessionActivitySummaries</code>	Sitzungszahlen und Anmeldedaten nach Zeitraum und Bereitstellungsgruppe
<code>http://{dc-host}/Citrix/Monitor/OData/v1/Data/Sessions</code>	Repräsentiert einen mit einem Desktop verbundenen Benutzer

URL	Beschreibung
http://{dc-host}/Citrix/Monitor/OData/v1/Data/TaskLogs	Protokoll aller Tasks, die als Teil des internen Überwachungsdiensts ausgeführt wurden, und von deren Status
http://{dc-host}/Citrix/Monitor/OData/v1/Data/Users	Benutzer, die eine Sitzung in der Site gestartet haben

SDK

Sep 29, 2015

Für XenApp und XenDesktop gibt es ein SDK, das auf zahlreichen Microsoft Windows PowerShell Version 3.0-Snap-Ins basiert, mit dem Sie die gleichen Tasks wie mit der Citrix Studio-Konsole und andere Tasks ausführen können, die in Studio allein nicht möglich sind.

Ab Version 7.5 bieten XenApp und XenDesktop mit FlexCast Management Architecture eine einheitliche Architektur und Verwaltung. Daher stehen nun in XenApp viele Features bereit, die es zuvor nur in XenDesktop gab. Elemente des SDKs für gemeinsame Features gelten also nun sowohl für XenApp als auch für XenDesktop, selbst wenn die Befehle sich nur auf XenDesktop beziehen.

Das SDK wurde in diesem Release wie folgt erweitert:

- Die PowerShell SDK-Cmdlets wurden erweitert, um Folgendes einzuschließen:
 - Die neuen unterstützten Hypervisortypen, einschließlich AWS und CloudPlatform
 - Unterstützung für Wake-On-LAN-Maschinen mit dem neuen Hypervisortyp "WakeOnLAN"
- Mit den Erweiterungen zum SDK können Sie Eigenschaften speziell für Cloudbereitstellungen konfigurieren, etwa die Region für die Verbindung und Volumeservice, sowie Faktoren steuern, wie die Konfiguration der Cloudfirewall, um einen Netzwerkausgang von virtuellen privaten Clouds zuzulassen.
- Im Zusammenhang mit der Unterstützung für Wake-On-LAN (WOL) können Kataloge für Remote-PC-Zugriff nun einer WOL-Hypervisorverbindungsinstanz zugeordnet werden und jede Maschine kann maschinenspezifische Energieverwaltungsfunktionen berichten.
- **Neues SDK auf höherer Ebene:** XenDesktop 7 stellt ein neues SDK auf höherer Ebene bereit, mit dem Sie schnell und mühelos ein Skript erstellen und die Siteerstellung und -wartung automatisieren. Das SDK auf höherer Ebene isoliert Sie von der Komplexität der SDKs auf niedrigerer Ebene, sodass Sie für das Erstellen einer neuen Site nur zwei Cmdlets ausführen müssen.
- **Neue SDKs auf niedriger Ebene:** Einzelne SDKs auf niedriger Ebene werden für die neuen XenDesktop 7-Dienste bereitgestellt, u. a. ein dediziertes und verbessertes SDK für den Dienst für die delegierte Administration (DAS), das vorher Teil des Broker-SDKs in XenDesktop 5 war. Es gibt auch SDKs für neue Features, u. a. Überwachungsdienst, Umgebungstest und Konfigurationsprotokollierung.
- **Kataloge und Bereitstellungsgruppen für Windows-Serverbetriebssystemmaschine:** Mit dem XenDesktop 7-SDK stellen Sie kostengünstig Anwendungen und Desktops bereit, die auf Serverbetriebssystemen gehostet werden.
- **Anwendungen für Desktopbetriebssystemmaschinen:** Anwendungen für Desktopbetriebssystemmaschinen haben sich auf der SDK-Ebene wesentlich geändert. Wenn Sie vorhandene Skripts für die Ausführung von Anwendungen auf Desktopbetriebssystemen haben, müssen Sie diese Skripts für XenDesktop 7 aktualisieren, da es fast keine Rückwärtskompatibilität gibt.
- **Anwenden von Einstellungen auf Maschinen in Bereitstellungsgruppen:** In XenDesktop 7 können Sie mit Konfigurationsslots Einstellungen auf Maschinen in einer bestimmten Bereitstellungsgruppe statt auf alle Maschinen in einer Site anwenden. So können Sie für eine bestimmte Bereitstellungsgruppe konfigurieren, welche Einstellungen für diese Gruppe gelten. Zahlreiche vordefinierte Konfigurationsslots werden bereitgestellt, die unterschiedliche Typen von Einstellungen enthalten, u. a. Einstellungen für StoreFront-Adressen, die mit Speicherorten von Receiver- oder App-V-Veröffentlichungserver verwendet werden. Eine Sammlung von Einstellungen kann sich nur auf eine bestimmte

Bereitstellungsgruppe auswirken, und eine andere Sammlung von Einstellungen von dem gleichen Slot wirkt sich auf eine andere Bereitstellungsgruppe aus. Sie können Namen verwenden, die für Ihre Bereitstellung geeignet sind, z. B. "Vertriebsabteilungsrichtlinie".

- **Ersetzte Katalogtypen:** In XenDesktop 7 wurden Katalogtypen durch Kataloge mit individuellen Eigenschaften ersetzt. Für die Rückwärtskompatibilität können Sie weiterhin vorhandene Skripts nutzen, die Katalogtypen verwenden, z. B. ein Image (gepoolt) und Thin Clone (dediziert) usw., intern werden diese jedoch in Gruppen von Eigenschaften konvertiert.

Achtung: Die Rückwärtskompatibilität mit XenDesktop 5-Katalogtypen wurde beibehalten, wenn es möglich und praktikabel war. Verwenden Sie beim Schreiben neuer Skripts jedoch keine Katalogtypen sondern geben Sie Kataloge mit individuellen Eigenschaften an.

- **Ersetzte Desktopobjekte:** In XenDesktop 5 ist das Desktopobjekt einer der Haupttypen des SDK-Objekts, das in Broker-SDK-Skripts verwendet wird. Das Desktopobjekt beschreibt sowohl die Maschine als auch die Sitzung auf der Maschine. In XenDesktop 7 wird dieses Objekt durch das Sitzungsobjekt und das Maschinenobjekt ersetzt, die beide erweitert wurden und jetzt die Funktion des Desktopobjekts übernehmen. Für die Rückwärtskompatibilität können Sie jedoch weiterhin vorhandene Skripts verwenden, die das Desktopobjekt nutzen.

Achtung: Die Rückwärtskompatibilität mit XenDesktop 5 wurde beibehalten, wenn es möglich und praktikabel war. Verwenden Sie beim Schreiben neuer Skripts jedoch nicht das Desktopobjekt, sondern geben Sie Sitzungs- und Maschinenobjekte an.

Die Richtlinienregeln im SDK und der Studio-Konsole sind nicht identisch. Anspruchs- und Zuordnungsrichtlinienregeln sind unabhängige Entitäten im SDK, und in der Konsole. Diese Entitäten sind nicht sichtbar, da sie nahtlos mit der Bereitstellungsgruppe zusammengeführt werden. Weiterhin sind Richtlinienregeln im SDK weniger restriktiv.

Das SDK enthält mehrere PowerShell-Snap-Ins, die automatisch vom Installationsassistenten installiert werden, wenn Sie den Controller oder Studio installieren.

So greifen Sie auf die Cmdlets zu:

1. Starten Sie eine Shell in PowerShell 3.0.

Klicken Sie zum Starten einer Shell über die Konsole auf **Studio**, wählen Sie die Registerkarte "PowerShell" aus und klicken Sie auf **PowerShell starten**.

Sie müssen die Shell oder das Skript mit einer ID ausführen, die über Citrix Administratorrechte verfügt. Obwohl die Mitglieder der lokalen Administratorgruppe auf dem Controller automatisch über Volladministratorprivilegien verfügen, um XenDesktop zu installieren, empfiehlt Citrix, dass Sie für den normalen Betrieb Citrix Administratoren mit den entsprechenden Rechten erstellen und nicht das lokale Administratorkonto verwenden. Wenn Sie Windows Server 2008 ausführen, müssen Sie die Shell oder das Skript als Citrix Administrator und nicht als Mitglied der lokalen Administratorgruppe ausführen.

2. Legen Sie die Ausführungsrichtlinie in PowerShell fest, um SDK-Cmdlets in Skripts zu verwenden.
Weitere Informationen zur PowerShell-Ausführungsrichtlinie finden Sie in der Dokumentation von Microsoft.
3. Fügen Sie mit dem Befehl **Add -PSSnapin** in der Windows PowerShell-Konsole die Snap-Ins hinzu, die Sie in der PowerShell-Umgebung benötigen. V1 und V2 geben die Version des Snap-Ins an (XenDesktop 5-Snap-Ins sind Version 1; XenDesktop 7-Snap-Ins sind Version 2). Geben Sie zum Beispiel Folgendes ein:
Add-PSSnapin Citrix.ADIdentity.Admin.V2

Geben Sie zum Importieren aller Cmdlets Folgendes ein:

```
Add-PSSnapin Citrix.*.Admin.V*
```

Nach dem Import haben Sie Zugriff auf die Cmdlets und die zugehörige Hilfe.

Ein Beispiel eines typischen Anwendungsfalls finden Sie unter [Erste Schritte mit dem SDK](#).

Tipp: Eine vollständige Liste der gesamten Hilfe für die Cmdlets finden Sie unter [Hilfe zu PowerShell-Cmdlets](#).

Mit dem Citrix Group Policy SDK können Sie Einstellungen und Filter für Gruppenrichtlinien anzeigen und konfigurieren. Es verwendet einen PowerShell-Anbieter, um einen virtuellen Datenträger zu erstellen, der mit den Maschinen- und Benutzereinstellungen und -filtern übereinstimmt. Der Anbieter wird als Erweiterung zu New-PSDrive angezeigt. Für die Verwendung des Group Policy SDKs muss Studio oder das XenApp- bzw. XenDesktop-SDK installiert sein.

Hinzufügen des Group Policy SDKs

1. Zum Hinzufügen des Gruppenrichtlinien-SDKs geben Sie Folgendes ein:

```
Add-PSSnapin citrix.common.grouppolicy
```

2. Geben Sie Folgendes ein, um auf die Hilfe zuzugreifen:

```
help New-PSDrive -path localgpo:/
```

Verwenden des Group Policy SDKs

1. Zum Erstellen eines virtuellen Laufwerks und zum Laden der Einstellungen geben Sie Folgendes ein:

```
New-PSDrive [-PSProvider] CitrixGroupPolicy
```

— *-Controller*

```
New-PSDrive [-PSProvider] CitrixGroupPolicy
```

— *-Controller*

wobei

— *-Controller*

der vollqualifizierte Domänenname eines Controllers in der Site ist, mit der Sie eine Verbindung herstellen und aus der Sie Einstellungen laden möchten.

Erste Schritte mit dem SDK

Sep 29, 2015

In diesen Abschnitten wird die Verwendung des SDKs in der Bereitstellung beschrieben. Es wird das empfohlene Verfahren zum Erstellen eines Skripts beschrieben. Sie können dieses Verfahren als Grundlage zur Erstellung von Skripten für andere Zwecke verwenden. Beispiele typischer Anwendungsfälle sind ebenfalls aufgeführt.

Erstellen eines Skripts

Führen Sie folgende Schritte für das Erstellen eines Skripts aus:

1. Führen Sie in Citrix Studio den Vorgang aus, für den Sie ein Skript erstellen möchten, z. B. Erstellen eines Katalogs für eine Gruppe von Maschinenerstellungsdienste-Maschinen.
2. Sammeln Sie das Protokoll der SDK-Vorgänge, die Studio für das Durchführen der Task ausgeführt hat.
3. Überprüfen Sie das Skript, um die Funktion jedes Teils zu verstehen. Dies hilft Ihnen bei der Anpassung des eigenen Skripts. Weitere Informationen finden Sie im Beispiel eines Anwendungsfalls, in dem die Skriptvorgänge ausführlich beschrieben werden.
4. Konvertieren und passen Sie das Studio-Skriptfragment an, um es in ein besser verwendbares Skript zu verwandeln. Gehen Sie hierzu folgendermaßen vor:
 - Verwenden Sie Variablen. Für einige Cmdlets können Parameter angegeben werden, z. B. TaskId. Allerdings kann es nicht eindeutig sein, woher der in diesen Parametern verwendete Wert stammt, da Studio Werte von den Ergebnisobjekten früherer Cmdlets verwendet.
 - Entfernen Sie alle Befehle, die nicht erforderlich sind.
 - Fügen Sie einige Schritte in einer Schleife hinzu, damit sie einfach zu steuern sind. Beispiel: Fügen Sie die Maschinenerstellung in einer Schleife hinzu, damit Sie die Anzahl der erstellten Maschinen steuern können.

Beispiele

Hinweis: Um beim Erstellen eines Skripts sicherzustellen, dass Sie immer die aktuellen Verbesserungen und Fixes erhalten, empfiehlt Citrix, dass Sie den oben erläuterten Schritten folgen, anstatt das Beispielskript zu kopieren und einzufügen.

Beispiele	Beschreibung
Create catalog	Skript: Erstellen eines Katalogs für eine Gruppe von Maschinenerstellungsdienste-Maschinen
Beispiel: Erstellen und Konfigurieren eines Hosts	Skript: Erstellen und Konfigurieren eines Hosts
Beispiel: Erstellen eines PvD-Desktops	Skript: Erstellen einer Bereitstellungsgruppe, die PvD-Desktops (Personal vDisk) enthält
Beispiel: Abrufen von Lastausgleichsinformationen	Anzeigen der Lastindexwerte für Serverbetriebssystemmaschinen

Beispiel: Erstellen eines Katalogs

Sep 29, 2015

Das folgende Beispiel zeigt, wie ein Katalog für eine Gruppe von Maschinenerstellungsdienste-Maschinen erstellt wird.

Bevor Sie beginnen, stellen Sie sicher, dass Sie die in **Erste Schritte mit dem SDK** aufgeführten Schritte ausführen. In diesem Abschnitt wird dargestellt, wie Sie Studio für die Durchführung des Vorgangs verwenden, für den Sie ein Skript erstellen möchten (in diesem Fall das Erstellen eines Katalogs für eine Gruppe von Maschinenerstellungsdienste-Maschinen) und das Sammeln des Protokolls der SDK-Vorgänge, die Studio für das Ausführung der Aufgabe vorgenommen hat. Die Ausgabe kann dann so angepasst werden, dass ein Skript für die Automatisierung der Katalogerstellung erstellt wird.

Hinweis: Um sicherzustellen, dass Sie immer die aktuellen Verbesserungen und Fixes erhalten, empfiehlt Citrix, dass Sie den in diesem Abschnitt beschriebenen Schritten folgen, anstatt das Beispielskript zu kopieren und einzufügen. Zeilennummern und Zeilenumbrüche wurden dem Skript für eine bessere Lesbarkeit hinzugefügt.

Inhalt des Skripts

Im folgenden Abschnitt wird die Funktion jedes Teils des Skripts erläutert, das Studio erstellt. Dies hilft Ihnen bei der Anpassung des eigenen Skripts. Zeilennummern wurden für eine bessere Lesbarkeit hinzugefügt.

1. **Start-LogHighLevelOperation** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -Source 'Studio' -StartTime 29/05/2013 14:43:08 -Text 'Create Machine Catalog "ExampleMachines"'
Startet einen protokollierten Vorgang und gibt eine Protokoll-ID zurück, die an nachfolgende Vorgänge übergeben wird, um sie der größeren Task zuzuordnen.
2. **New-BrokerCatalog** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -AllocationType 'Permanent' -Description 'Example Machines' -IsRemotePC \$False -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46
Erstellt einen Brokerkatalog. Dieser Katalog enthält Maschinen, die Sie gerade erstellen.
3. **New-AcctIdentityPool** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -AllowUnicode -Domain 'dumdev.internal.citrix.com' -IdentityPoolName 'ExampleMachines' -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46
Definiert den Mechanismus zum Erstellen von Active Directory-Computerkonten. Dies wird zu einem Container für Active Directory-Konten, die für die Maschinen erstellt werden, die Sie erstellen.
4. **Set-BrokerCatalogMetadata** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -CatalogId 1 -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46 -Name 'Citrix_DesktopStudio_IdentityPoolUid' -Value 'f39a2792-064a-43eb-97c7-397cc1238e46'
Stellt Metadaten im Brokerkatalog ein, die Details des Identitätspool enthalten. Dies ist nicht unbedingt erforderlich.
5. **Test-ProvSchemeNameAvailable** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -ProvisioningSchemeName @('ExampleMachines')
Überprüft, ob der gewünschte Name verfügbar ist. Dies ist nicht unbedingt erforderlich.
6. **New-ProvScheme** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -CleanOnBoot -HostingUnitName 'SharedNFS' -IdentityPoolName 'ExampleMachines' -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46
Erstellt ein Bereitstellungsschema-Objekt. Dies ist eine Vorlage für die Maschinen, die Sie erstellen möchten. Sie gibt den Hypervisor, das Netzwerk, den Speicher, den Speicherplatz, die Anzahl der CPUs usw. an. Sie übernimmt Parameter vom bereits eingerichteten System, u. a. den Namen des Hostingeräts und den Pfad zum VM-Snapshot, die für die Maschinen verwendet werden, die Sie erstellen. Dieser Befehl erstellt eine konsolidierte Kopie des verwendeten VM-Snapshots, daher kann der Abschluss des Prozesses länger dauern.

In diesem Beispiel gibt das Studio-Skript das Flag -RunAsynchronous in diesem Befehl an. Dies bedeutet, dass der Befehl vor dem Abschluss die Steuerung an den Administrator zurückgibt; Sie müssen daher auf den Abschluss warten, bevor Sie Vorgänge durchführen, für die der Befehl abgeschlossen sein muss. Wenn dieses Flag nicht angegeben ist, wird der Befehl synchron intern ausgeführt und die Steuerung wird nicht zurückgegeben, bis der Befehl abgeschlossen ist (erfolgreich oder fehlerhaft). Sie können den Status einer asynchronen Task mit dem Cmdlet Get-ProvTask überprüfen. Geben Sie die Task-ID an, die der Vorgang zurückgegeben hat, der die Task gestartet hat, in diesem Fall das Cmdlet New-ProvScheme.

7. **Set-BrokerCatalog** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46 -Name 'ExampleMachines' -ProvisioningSchemeId 76125e3a-9001-4993-8000-000000000000
Aktualisiert den Brokerkatalog mit der eindeutigen ID des oben erstellten Provisioningschemas.
8. **Add-ProvSchemeControllerAddress** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -ControllerAddress @('DDC.dumdev.internal.citrix.com') -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46
Fügt dem Provisioningschema-Objekt eine Gruppe von Controlleradressen hinzu. Dies ist eine Liste von Adressen, die die erstellten Maschinen für die Registrierung bei einem Controller (Broker) verwenden können, wenn sie bereitgestellt werden. Die Registrierungsadressen der Maschinen können mit mehreren Methoden angegeben werden. Diese Informationen sind jedoch erforderlich, wenn der Administrator im VDA-Installationsprogramm die Adressen mit den Maschinenerstellungsdiensten angeben möchte. Änderungen an dieser Liste wirken sich nur auf Maschinen aus, die nach der Änderung erstellt werden, nicht auf vorhandene Maschinen.
9. **Get-AcctADAccount** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -IdentityPoolId b99aee6d-8772-4dbc-978b-8eb9a26e2407 -Lock \$False -MaxRecordCount 2147483647 -State 'Available'
Studio erhält eine Liste der verfügbaren Maschinen-IDs vom Identitätspool, damit vorhandene Konten, die nicht verwendet werden, verbraucht werden können statt neue Konten zu erstellen. Hinweis: Dies wird nicht im Skript benötigt, da neue Konten erstellt werden können, wenn das Skript in einem Kontext ausgeführt wird, der entsprechende Berechtigungen hat. Wenn das Skript jedoch keine Berechtigungen zum Erstellen von Konten hat, ändern Sie das Skript so, dass verfügbare Konten verbraucht werden (ein eigenständiger Prozess wird benötigt, um ein Pool der Konten dem Identitätspool bereitzustellen, bevor das Skript ausgeführt wird).
10. **New-AcctADAccount** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -Count 2 -IdentityPoolId b99aee6d-8772-4dbc-978b-8eb9a26e2407 -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46
Erstellt die erforderlichen Active Directory-Computerkonten in Active Directory. Das Skript erstellt ein Konto und kann, falls erforderlich, weitere Konten mit dem Parameter "Count" des Befehls erstellen. Die Konten werden in der Organisationseinheit erstellt, die im obigen Provisioningschema festgelegt wurde.
11. **New-ProvVM** -ADAccountName @('DUMDEV\Example-0001\$', 'DUMDEV\Example-0002\$') -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46
Erstellt virtuelle Maschinen basierend auf der Vorlagendefinition im oben erstellten Provisioningschema. Der Abschluss dieses Prozesses kann länger dauern.
12. **Lock-ProvVM** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46 -ProvisioningSchemeName 'ExampleMachines' -Tag 'Brokered' -VMID @('0710bb77-d01f-d006-4d67-5472e5cd349f')
Sperrt die bereitgestellten virtuellen Maschinen und verhindert versehentliche Modifikationen an der virtuellen Maschine. Consumer des SDKs können hiermit angeben, dass die virtuelle Maschine verwendet wird und den Grund für die Sperrung. Das Skript sperrt die VM mit einem Tag "Brokered". Dies gibt an, dass die virtuelle Maschine erstellt und dem Brokerkatalog hinzugefügt wurde und erst nach dem Entfernen aus dem Katalog gelöscht werden kann. Sie können den Tagnamen beliebig angeben.
13. **New-BrokerMachine** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -CatalogId 1 -HostedMachineId '0710bb77-d01f-d006-4d67-5472e5cd349f' -HypervisorConnectionId 1 -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46
Erstellt ein Broker-Maschinenobjekt. Dies sind im Katalog gespeicherte Objekte, die die bereitgestellte Maschine mit dem Katalog zusammenführen.
14. **Start-BrokerMachinePvdImagePrepare** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -InputObject @(2) -LoggingId f39a2792-064a-43eb-97c7-397cc1238e46
Fordert den Brokerdienst auf, einen Vorbereitungsvorgang für Personal vDisk auszulösen. Dies ist erforderlich, damit die Maschine den Speicher für Personal vDisk initialisieren kann.
15. **Stop-LogHighLevelOperation** -AdminAddress 'ddc.dumdev.internal.citrix.com:80' -HighLevelOperationId f39a2792-064a-43eb-97c7-397cc1238e46 -IsSuccessful \$true
Beendet die Protokollierung, die im ersten Schritt begonnen wurde, und gibt einen Erfolg an.

Anpassen des Skripts

Im folgenden Abschnitt wird erläutert, wie Sie die Studio-Ausgabe in ein Skript konvertieren und anpassen, das besser verwendbar ist. Zusätzlich zur Verwendung von Variablen und dem Entfernen von nicht benötigten Befehlen erfahren Sie, wie Sie die Maschinenerstellung einer Schleife hinzufügen, damit Sie die Anzahl der erstellten Maschinen steuern können. Zeilennummern wurden für eine bessere Lesbarkeit hinzugefügt.

```
1. [CmdletBinding()] param ( [Parameter(Mandatory=$true)] [string] $hostingUnitPath, [Parameter(Mandatory=$true)] [string] $catalogName, [string] $catalogDescription, [Parameter(Mandatory=$true)] [int]
```

Beispiel: Erstellen und Konfigurieren eines Hosts

Sep 29, 2015

Das folgende Beispiel zeigt das Erstellen und Konfigurieren eines Hosts.

Bevor Sie beginnen, stellen Sie sicher, dass Sie die in [Erste Schritte mit dem SDK](#) aufgeführten Schritte ausführen. In diesem Abschnitt wird dargestellt, wie Sie Studio für die Durchführung des Vorgangs verwenden, für den Sie ein Skript erstellen möchten (in diesem Fall das Erstellen eines Hosts), und für das Sammeln des Protokolls der SDK-Vorgänge, die Studio für die Ausführung der Aufgabe vorgenommen hat. Die Ausgabe kann dann so angepasst werden, dass ein Skript für die Automatisierung der Hosterstellung erstellt wird.

Hinweis: Um sicherzustellen, dass Sie immer die aktuellen Verbesserungen und Fixes erhalten, empfiehlt Citrix, dass Sie den in diesem Abschnitt beschriebenen Schritten folgen, anstatt das Beispielskript zu kopieren und einzufügen. Zeilennummern und Zeilenumbrüche wurden dem Skript für eine bessere Lesbarkeit hinzugefügt.

Inhalt des Skripts

Im folgenden Abschnitt wird die Funktion jedes Teils des Skripts erläutert, das Studio erstellt. Dies hilft Ihnen bei der Anpassung des eigenen Skripts. Zeilennummern wurden für eine bessere Lesbarkeit hinzugefügt.

1. Get-LogSite -AdminAddress 'mycontroller.example.com:80'

Führt eine Abfrage beim Konfigurationsprotokollierungsdienst durch, um Informationen zur Sitekonfiguration zu erhalten.

2. Start-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -Source 'Studio' -StartTime 14/08/2013 14:30:28 -Text 'Create Connection "Example XenServer"'

Startet einen allgemeinen Protokollierungsvorgang bei der Konfigurationsprotokollierung, in der die restlichen Befehle bestehen. Gibt eine Protokoll-ID zurück, die an nachfolgende Vorgänge übergeben wird.

3. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

Legt den Speicherort des Hostdienstes fest, der von den Konfigurations-Cmdlets verwendet wird. Da der Hostdienst einen PowerShell-Anbieter offenlegt, können nicht alle Cmdlets eine Adresse für den Dienst annehmen. Daher legt dieses Cmdlet einen Standardspeicherort fest.

4. New-Item -ConnectionType 'XenServer' -HypervisorAddress @('http://xenhost1.example.com') -LoggingId e355ce51-8cbb-400a-ae81-1fdc567239cb -Path @('XDHyp:\Connections\Example XenServer')
Erstellt eine Verbindung zu XenServer (xenhost1.beispiel.com). Dies ist eine nicht-persistente Verbindung und ist nur für diesen PowerShell-Runspace verfügbar.

5. Stop-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -EndTime 14/08/2013 14:30:29 -HighLevelOperationId 'e355ce51-8cbb-400a-ae81-1fdc567239cb' -IsSuccessful \$True
Beendet die zuvor begonnene Protokollierung und gibt einen Erfolg an.

6. Get-LogSite -AdminAddress 'mycontroller.example.com:80'

Führt eine Abfrage beim Konfigurationsprotokollierungsdienst durch, um Informationen zur Sitekonfiguration zu erhalten.

7. Start-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -Source 'Studio' -StartTime 14/08/2013 14:30:30 -Text 'Update Connection "Example XenServer"'

Startet einen neuen allgemeinen Protokollierungsvorgang.

8. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

Legt erneut die Adressangaben des Hostdienstes fest (diese Wiederholung wurde in dem optimierten Skript unten entfernt).

9. Set-Item -HypervisorAddress @('http://xenhost1.example.com','http://xenhost2.example.com') -LoggingId 44e15629-6906-4840-a36c-984aaf67be6d -PassThru -Path @('XDHyp:\Connections\Example XenServer')
Aktualisiert die in Schritt 4 erstellte Verbindung. Da sich mehr als ein XenServer im Pool befindet, werden alle Adressen angegeben, damit hohe Verfügbarkeit möglich ist.

10. Stop-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -EndTime 14/08/2013 14:30:31 -HighLevelOperationId '44e15629-6906-4840-a36c-984aaf67be6d' -IsSuccessful \$True
Beendet den in Schritt 7 begonnenen Protokollierungsvorgang.

11. Get-LogSite -AdminAddress 'mycontroller.example.com:80'

Führt eine Abfrage beim Konfigurationsprotokollierungsdienst durch, um Informationen zur Sitekonfiguration zu erhalten.

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'

Startet einen neuen Protokollierungsvorgang.

13. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

Legt erneut die Adressangaben des Hostdienstes fest.

14. Get-ChildItem -Path @('XDHyp:\Connections')

Füllt die Dialogfelder des Assistenten mit den Inhalten der Hostverbindung aus.

15. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

Legt erneut die Adressangaben des Hostdienstes fest.

16. Remove-Item -LoggingId 76caa3f4-df93-4cb2-b78d-6a8824766314 -Path @('XDHyp:\Connections\Example XenServer')

Entfernt die im Assistenten erstellte temporäre Verbindung.

17. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

Legt erneut die Adressangaben des Hostdienstes fest.

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')
Erstellt die Verbindung erneut als persistente Verbindung, die in die Datenbank geschrieben wird und somit anderen PowerShell-Runspace zur Verfügung steht.

19. New-BrokerHypervisorConnection -AdminAddress 'mycontroller.example.com:80' -HypHypervisorConnectionUid a14096ba-5074-44ff-b596-371e345c0449 -LoggingId 76caa3f4-df93-4cb2-b78d-6a8824766314
Fügt die Hostverbindung zum Brokerdienst hinzu.

20. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

Legt erneut die Adressangaben des Hostdienstes fest.

21. New-Item -HypervisorConnectionName 'Example XenServer' -LoggingId 76caa3f4-df93-4cb2-b78d-6a8824766314 -NetworkPath @('XDHyp:\Connections\Example XenServer\Network 0.network') -Path @('XDHyp:\Connections\Example XenServer\Network 0.network')
Erstellt die HostingUnit (in Studio als "Ressourcen" bezeichnet) mit den in Schritt 14 gesammelten Informationen.

22. Set-HypAdminConnection -AdminAddress 'mycontroller.example.com:80'

Legt erneut die Adressangaben des Hostdienstes fest.

23. Get-Item -Path @(XDHyp:\Connections\Example XenServer)
Ruft ein neu erstelltes Objekt ab.

Stop-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -EndTime 14/08/2013 14:31:07 -HighLevelOperationId '76caa3f4-df93-4cb2-b78d-6a8824766314' -IsSuccessful \$True
Beendet die zuvor begonnene Protokollierung und gibt an, ob sie erfolgreich war.

Anpassen des Skripts

Im folgenden Abschnitt wird erläutert, wie Sie die Studio-Ausgabe in ein Skript konvertieren und anpassen, das besser verwendbar ist. Das folgende Skript wurde vereinfacht, sodass anstelle der temporären Hostverbindung, die mit dem Studio-Skript erstellt wurde, um die Assistenten mit Informationen zu versorgen, eine persistente Verbindung erstellt wird. Aus dieser Verbindung heraus werden dann Informationen für die Erstellung der HostingUnit (Ressourcen) abgefragt. Hinweis: Die LoggingId- und HypHyperConnectionUid-Details sind unterschiedlich.

Für bessere Lesbarkeit wurden Zeilennummern hinzugefügt. Jedes nummerierte Element steht für einen PowerShell-Befehl.

1. Start-LogHighLevelOperation -AdminAddress 'mycontroller.example.com:80' -Source 'Studio' -Text 'Create Connection "Example XenServer"' 2. Set-HypAdminConnection -AdminAddress 'mycontrolle

Beispiel: Erstellen eines PvD-Desktops

Sep 29, 2015

Mit dem Skriptbeispiel in diesem Abschnitt kann eine Bereitstellungsgruppe erstellt werden, die PvD-Desktops (Personal vDisk) enthält.

Bevor Sie beginnen, stellen Sie sicher, dass Sie die in [Erste Schritte mit dem SDK](#) aufgeführten Schritte ausführen. In diesem Abschnitt wird dargestellt, wie Sie Studio für die Durchführung des Vorgangs verwenden, für den Sie ein Skript erstellen möchten, und für das Sammeln des Protokolls der SDK-Vorgänge, die Studio für die Ausführung der Aufgabe vorgenommen hat. Die Ausgabe kann so angepasst werden, dass ein Skript für die Automatisierung der Aufgabe erstellt wird.

Hinweis: Um sicherzustellen, dass Sie immer die aktuellen Verbesserungen und Fixes erhalten, empfiehlt Citrix, dass Sie den in diesem Abschnitt beschriebenen Schritten folgen, anstatt das Beispielskript zu kopieren und einzufügen.

Inhalt des Skripts

Im folgenden Abschnitt wird die Funktion jedes Teils des Skripts erläutert, das Studio erstellt. Dies hilft Ihnen bei der Anpassung des eigenen Skripts. Zeilennummern und Zeilenumbrüche wurden dem Skript für eine bessere Lesbarkeit hinzugefügt.

- 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''**
Startet einen protokollierten Vorgang und gibt eine Protokoll-ID zurück, die an nachfolgende Vorgänge übergeben wird, um sie der größeren Aufgabe zuzuordnen.
- 2. New-BrokerDesktopGroup -AdminAddress 'test-ddc.mydomain.com:80' -ColorDepth 'TwentyFourBit' -DeliveryType 'DesktopsOnly' -DesktopKind 'Private' -InMaintenanceMode \$False -IsRemotePC \$False**
Erstellt eine neue Bereitstellungsgruppe mit Optionen, die vom Studio-Assistenten gesammelt wurden.
- 3. Add-BrokerMachinesToDesktopGroup -AdminAddress 'test-ddc.mydomain.com:80' -Catalog 'win7-pvd' -Count 2 -DesktopGroup 'Win7 PvD Desktops' -LoggingId 846f2d42-a994-4bce-ab58-be05c8d73b99'**
Fügt die vom benannten Katalog angeforderte Anzahl an Maschinen zur neuen Bereitstellungsgruppe hinzu.
- 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'**
Die oben aufgeführten Befehle sind nicht erforderlich, da Benutzer von Studio verifiziert werden.
- 5. Test-BrokerAssignmentPolicyRuleNameAvailable -AdminAddress 'test-ddc.mydomain.com:80' -Name @('Win7 PvD Desktops')**
Studio prüft, ob der Richtlinienzuordnungsname zur Verfügung steht.
- 6. New-BrokerAssignmentPolicyRule -AdminAddress 'test-ddc.mydomain.com:80' -DesktopGroupUid 41 -Enabled \$True -IncludedUserFilterEnabled \$False -LoggingId 846f2d42-a994-4bce-ab58-be05c8d73b99'**
Erstellt die neue Richtlinienzuordnungsregel für die Bereitstellungsgruppe. Es sind keine Benutzer angegeben, daher erfolgt die Steuerung durch die Zugriffsrichtlinienregel.
- 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'**
Die oben aufgeführten Befehle sind nicht erforderlich, da Studio weitere Prüfungen durchführt.
- 8. Test-BrokerAccessPolicyRuleNameAvailable -AdminAddress 'test-ddc.mydomain.com:80' -Name @('Win7 PvD Desktops_Direct')**
Studio stellt sicher, dass der Regelname für die Zugriffsrichtlinie zur Verfügung steht.
- 9. New-BrokerAccessPolicyRule -AdminAddress 'test-ddc.mydomain.com:80' -AllowedConnections 'NotViaAG' -AllowedProtocols @('HDX'; 'RDP') -AllowRestart \$True -DesktopGroupUid 41 -Enabled \$True**
Erstellt die Zugriffsrichtlinienregel für den neuen Desktop für Verbindungen, die nicht über NetScaler Gateway erfolgen.
- 10. Test-BrokerAccessPolicyRuleNameAvailable -AdminAddress 'test-ddc.mydomain.com:80' -Name @('Win7 PvD Desktops_AG') New-BrokerAccessPolicyRule -AdminAddress 'test-ddc.mydomain.com:80'**
Studio wiederholt diesen Vorgang für Verbindungen über NetScaler Gateway.
- 11. Test-BrokerPowerTimeSchemeNameAvailable -AdminAddress 'test-ddc.mydomain.com:80' -Name @('Win7 PvD Desktops_Weekdays') New-BrokerPowerTimeScheme -AdminAddress 'test-ddc.mydomain.com:80'**
Studio prüft, ob die Namen für die (optionalen) Wochentag- und Wochenend-Energieschemas verfügbar sind und fügt sie hinzu.
- 12. Stop-LogHighLevelOperation -AdminAddress 'test-ddc.mydomain.com:80' -EndTime 31/07/2013 10:09:05 -HighLevelOperationId '846f2d42-a994-4bce-ab58-be05c8d73b99' -IsSuccessful \$True**
Beendet die Protokollierung, die in Schritt 1 begonnen wurde, und gibt einen Erfolg an.

Anpassen des Skripts

In diesem Abschnitt wird erläutert, wie Sie die Studio-Ausgabe in ein besser verwendbares Skript konvertieren und anpassen.

Das Skript erstellt eine Bereitstellungsgruppe, die PvD-Desktops enthält. Der in den Parametern angegebene Katalog muss bereits vorhanden sein und die richtigen Daten enthalten (den Zuteilungstyp "Static" und einen PvD-Datenträger). Das Skript ist für die Ausführung von einer PowerShell-Befehlszeile konzipiert, wobei Sie als Citrix Administrator angemeldet sein müssen. Es werden keine Berechtigungen überprüft. Das Skript schlägt fehl, wenn der Benutzer nicht über die entsprechenden Berechtigungen verfügt.

<# Sample usage: .\CreatePvDGroup.ps1 -GroupName "Win7 PvD Desktops" -SrcCatalog "win7-pvd" -NumDesktops 2 -Users @('mydomain\user1'; 'mydomain\user2') -#> Param([Parameter(Mandatory=\$true)] \$GroupName, [Parameter(Mandatory=\$true)] \$SrcCatalog, [Parameter(Mandatory=\$true)] \$NumDesktops, [Parameter(Mandatory=\$true)] \$Users)

In der Tabelle werden die im Skript verwendeten Parameter erläutert.

Parameter	Beschreibung
GroupName	Der Name der neuen PvD-Desktopgruppe.
SrcCatalog	Der Name des Katalogs, der zum Erstellen des PvD-Desktops verwendet wird. Verwenden Sie beim Erstellen des Katalogs den Zuteilungstyp "Static". Auch Maschinen müssen PvD-Datenträger haben.
NumDesktops	Die Anzahl der Maschinen, die der PvD-Desktopgruppe hinzugefügt werden. Wenn nicht genug Maschinen verfügbar sind, werden so viele wie möglich hinzugefügt.
Benutzer	Die Benutzer, die auf die Gruppe zugreifen können. Dies ist eine Liste von Benutzern oder Gruppen, z. B. @('mydomain\Domain Users') oder @('mydomain\user1'; 'mydomain\user2')

Set-HypAdminConnection -AdminAddress \$adminAddress

Gibt die zu verwendende Hypervisor-Adminverbindung an. Damit entfällt für einige Befehle der Bedarf von -AdminAddress.

\$peakPoolSize = 2 \$weekendPoolSizeByHour = new-object int[] 24 \$weekdayPoolSizeByHour = new-object int[] 24 9..17 | % { \$weekdayPoolSizeByHour[\$_] = \$peakPoolSize } \$peakHours = (0..23 | % {
Hiernit werden 24 Elementarrays mit einer 1 oder einer 0 in jedem Eintrag erstellt. Auf diese Weise können Sie für die Bereitstellungsgruppen die Spitzenzeiten für die Energiezeitpläne festlegen. Die

Elemente 9 bis 17 (Stunden beginnend von 09:00 bis 17:00) für Wochentage sind auf 1 eingestellt, während andere auf 0 festgelegt sind. Während der Spitzenzeiten werden zwei nicht zugewiesene Maschinen gestartet (wenn verfügbar).

```
$logId = Start-LogHighLevelOperation -Text "Create PvD desktop group" -Source "Create PvD Desktop Group Script"
```

Start einer neuen Protokollierung. Hierbei wird eine Protokoll-ID zurückgegeben, die an nachfolgende Vorgänge übergeben wird, um sie der Aufgabe zum Erstellen von Gruppen zuzuordnen.

```
$grp = New-BrokerDesktopGroup -DesktopKind 'Private' -DeliveryType 'DesktopsOnly' -LoggingId $logId -Name $GroupName -PublishedName $GroupName -SessionSupport 'SingleSession'
```

Erstellt die neue Bereitstellungsgruppe und stellt private Desktops bereit. Der verwendete Katalog muss passende Maschinen enthalten (permanent mit einem PvD-Datenträger). PublishedName ist der Name, der Endbenutzern angezeigt wird. Im Folgenden wird der Name als Gruppenname verwendet.

```
New-BrokerAssignmentPolicyRule -DesktopGroupUid $grp.Uid -IncludedUserFilterEnabled $False -LoggingId $logId -MaxDesktops 1 -Name ($GroupName + '_AssignRule') | Out-Null
```

Zugewiesene Desktops benötigen eine Zuordnungsrichtlinie. Deaktivieren Sie den Benutzerfilter, damit der Zugriff vollständig von Zugriffsrichtlinienregeln gesteuert wird.

```
New-BrokerAccessPolicyRule -AllowedConnections 'NotViaAG' -AllowedProtocols @('HDX','RDP') -AllowRestart $True -DesktopGroupUid $grp.Uid -IncludedSmartAccessFilterEnabled $True -Name ($GroupName + '_AccessRule')
```

Geben Sie Zugriffsbeschränkungen an: Ermöglichen Sie direkten Zugriff über NetScaler Gateway mit HDX- und RDP-Protokollen. Der Benutzer kann bei Bedarf einen Neustart des Desktops anfordern.

```
New-BrokerPowerTimeScheme -DaysOfWeek 'Weekdays' -DesktopGroupUid $grp.Uid -DisplayName 'Weekdays' -LoggingId $logId -Name ($GroupName + '_Weekdays') -PeakHours $peakHours
```

Optional: Geben Sie Energiezeitpläne an.

```
Stop-LogHighLevelOperation -HighLevelOperationId $logId -IsSuccessful $True
```

Beendet die Konfigurationsprotokollierung und gibt an, ob der Vorgang erfolgreich war oder nicht.

Beispiel: Abrufen von Lastausgleichinformationen

Sep 29, 2015

Sie können mit Serverbetriebssystemmaschinen kostengünstig auf Serverbetriebssystemen gehostete Anwendungen und Desktops mehreren Benutzern bereitstellen.

Der Lastausgleich für Serverbetriebssystemmaschinen in einer Bereitstellung erfolgt mit Citrix Richtlinien. Es gibt mehrere Load Balancing-Richtlinien zum Aktivieren und Konfigurieren der Lastverwaltung zwischen Servern, die Windows-Serverbetriebssystemmaschinen bereitstellen. Weitere Informationen zu diesen Richtlinien finden Sie unter [Einstellungen der Richtlinie "Lastverwaltung"](#). Sie arbeiten mit Richtlinien in Studio oder in der Gruppenrichtlinien-Verwaltungskonsolle in Windows. Weitere Informationen zum Arbeiten mit Richtlinien finden Sie unter [Verwalten von Citrix Richtlinien](#).

Sie zeigen die Last entweder mit Citrix Director oder mit den Studio-Konsolen oder mit dem PowerShell SDK an. Im folgenden Beispiel wird beschrieben, wie Sie die Last mit dem PowerShell SDK anzeigen.

Hinweis: Wenn Sie frühere Versionen von XenDesktop verwendet haben, sind Sie sicherlich mit dem Befehl `qfarm/load` vertraut. Dieses Tool steht nicht mehr zur Verfügung. Mit PowerShell können Sie jedoch eine ähnliche Ausgabe anzeigen, wie im Beispiel unten dargestellt.

Beispiel: Abrufen von Lastindexwerten

Folgen Sie den Anweisungen unten zum Anzeigen einer Liste von Maschinen mit den berechneten bzw. gemessenen Lastindexwerten und der Anzahl der auf den Maschinen ausgeführten Sitzungen:

1. Starten Sie eine Shell in PowerShell. Weitere Informationen finden Sie unter [Informationen zum XenApp- und XenDesktop-SDK](#).
2. Geben Sie Folgendes ein:
`Get-BrokerMachine -SessionSupport MultiSession -Property 'DnsName','LoadIndex','SessionCount'`

Weitere Informationen und Beispiele finden Sie in der Cmdlet-Hilfe für das Cmdlet `get-brokmachine` und in den Info-Abschnitten, u. a. [about_Broker_filtering-xd7.html](#). Siehe [Hilfe zu PowerShell-Cmdlets](#).

Hilfe zu PowerShell-Cmdlets

Sep 29, 2015

Dieser Abschnitt enthält die PowerShell-Hilfe für alle Cmdlets.

- 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

Sep 29, 2015

Name	Description
AcctAdIdentitySnapin	The Active Directory Identity Service PowerShell snap-in provides
Acct Filtering	Describes the common filtering options for XenDesktop cmdlets.

Name	Description
Add-AcctADAccount	Import Active Directory computer accounts from Active Directory for use in the AD Identity Service.
Add-AcctIdentityPoolScope	Add the specified IdentityPool(s) to the given scope(s).
Copy-AcctIdentityPool	Copies an Identity Pool and its associated Identities to a new IdentityPool
Get-AcctADAccount	Gets the AD accounts stored in the AD Identity Service.
Get-AcctDBConnection	Gets the database string for the specified data store used by the AdIdentity Service.
Get-AcctDBSchema	Gets a script that creates the AdIdentity Service database schema for the specified data store.
Get-AcctDBVersionChangeScript	Gets a script that updates the AdIdentity Service database schema.
Get-AcctIdentityPool	Gets existing identity pools.
Get-AcctInstalledDBVersion	Gets a list of all available database schema versions for the AdIdentity Service.
Get-AcctScopedObject	Gets the details of the scoped objects for the AdIdentity Service.
Get-AcctService	Gets the service record entries for the AdIdentity Service.
Get-AcctServiceAddedCapability	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.

Name	Description
Unlock-AcctIdentityPool	Unlocks identity pools.
Update-AcctADAccount	Refreshes the AD computer account state stored in the AD Identity Service.

about_AcctAdIdentitySnapin

Sep 29, 2015

about_AcctADIdentityServiceSnapin

The Active Directory Identity Service PowerShell snap-in provides administrative functions for the Active Directory Identity Service.

All commands in this snap-in have 'Acct' in their name.

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'

operations are also responsible for unlocking the identity pool. Use the unlock-AcctIdentityPool command to allow the lock to be overridden, if necessary.

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

Sep 29, 2015

XenDesktop - Advanced Dataset Filtering

Describes the common filtering options for XenDesktop cmdlets.

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.

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> ::= <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

Sep 29, 2015
Import Active Directory computer accounts from Active Directory for use in the AD Identity Service.

```
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>]
```

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.

- New-AcctADAccount
- Remove-AcctADAccount
- Repair-AcctADAccount
- Get-AcctADAccount
- Update-AcctADAccount
- Unlock-AcctADAccount

-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?	

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

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).

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.

----- 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

Sep 29, 2015

Add the specified IdentityPool(s) to the given scope(s).

```
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>]
```

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.

[Remove-AcctIdentityPoolScope](#)

[Get-AcctScopedObject](#)

-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

None

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.

----- 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

Sep 29, 2015

Copies an Identity Pool and its associated Identities to a new IdentityPool

Copy-AcctIdentityPool [-IdentityPoolName] <String> [-NewIdentityPoolName] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Copy-AcctIdentityPool -IdentityPoolUid <Guid> [-NewIdentityPoolName] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

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.

[New-AcctIdentityPool](#)

[Get-AcctIdentityPool](#)

[Remove-AcctIdentityPool](#)

-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

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.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.

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

Sep 29, 2015

Gets the AD accounts stored in the AD Identity Service.

```
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>]
```

Provides the ability to locate the AD accounts stored within the AD Identity Service and view the state of the accounts.

[New-AcctADAccount](#)

[Add-AcctADAccount](#)

[Remove-AcctADAccount](#)

[Unlock-AcctADAccount](#)

[Update-AcctADAccount](#)

[Repair-AcctADAccount](#)

-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

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.

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.

----- 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

Sep 29, 2015

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

Sep 29, 2015

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

-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 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

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-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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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
```

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-AcctService

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

-ADAccountName<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

Sep 29, 2015

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.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

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

Sep 29, 2015
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.AdIdentity.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 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

Sep 29, 2015

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

Sep 29, 2015
Removes metadata from the given IdentityPool.

Syntax

```
Remove-AcctIdentityPoolMetadata [-IdentityPoolUid] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Remove-AcctIdentityPoolMetadata [-IdentityPoolUid] <Guid> -Map <PSObject> [-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)

-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-AcctIdentityPool | % { Remove-AcctIdentityPoolMetadata -Map $_.MetadataMap }  
Remove all metadata from all IdentityPool objects.
```

Remove-AcctIdentityPoolScope

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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-AcctADAccout 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

Sep 29, 2015
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	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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-AcctIdentityPoolMetadata [-InputObject] <IdentityPool[]> -Map <PSObject> [-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?	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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015

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 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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

Cmdlets

Name	Description
ConvertTo-CtxAppVLauncherArg	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.
Get-CtxAppVApplication	Enumerates all published App-V applications for a given Management server.
Get-CtxAppVApplicationInfo	Enumerates information for a given application in a given package for a given Management server.
Get-CtxAppVServer	Returns URLs for App-V Publishing and Management servers contained in a Citrix App-V policy. Returned values are in string format.
Get-CtxAppVServerSetting	Returns settings for the specified App-V Publishing Server.
New-CtxAppVServer	Creates a new Citrix App-V policy containing the specified App-V Management and Publishing Server URLs.
Set-CtxAppVServerSetting	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.
Test-CtxAppVServer	Tests the given URL for the presence of App-V Management and Publishing servers.

ConvertTo-CtxAppVLauncherArg

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015
Enumerates information for a given application in a given package for a given Management server.

Syntax

```
Get-CtxAppVApplicationInfo [-AppVManagementServer] <string> [-Appld] <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)

-Appld<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" -Appld "[[ProgramFilesX86]]Beyond Compare 3\BCompare.exe" -PackageId "1bcb6993-10b1-4659-b9d4-e809e10cecdf_5"
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" -Appld "[[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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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 <String> -UserRefreshInterval <int> -UserRefreshIntervalUnit <int> -GlobalRefreshEnabled [<Boolean>] -GlobalRefreshOnLogon <String> -GlobalRefreshInterval <String> -GlobalRefreshIntervalUnit <String> [<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?	false
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?	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 5.0 documentation 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 5.0 documentation 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 5.0 documentation 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 5.0 documentation 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 5.0 documentation 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 5.0 documentation 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. Please refer to App-V 5.0 documentation for details. <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). Please refer to App-V 5.0 documentation for details. <http://technet.microsoft.com/en-us/library/jj687745.aspx>

Required?	true
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

Sep 29, 2015
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 <String> -UserRefreshInterval <int> -UserRefreshIntervalUnit <int> -GlobalRefreshEnabled [<Boolean>] -GlobalRefreshOnLogon <String> -GlobalRefreshInterval <String> -GlobalRefreshIntervalUnit <String> [<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

Sep 29, 2015
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

Sep 29, 2015
Overview

Name	Description
Broker AccessPolicy	Controls client-connection-based access to desktop groups.
Broker Applications	Describes how to publish and manage hosted applications.
Broker AssignmentPolicy	Controls the automatic, permanent assignment of machines to users.
Broker Concepts	Overview of the Citrix Broker.
Broker ConfigurationSlots	Overview of assigning a collection of related settings to a desktop group.
Broker ControllerDiscovery	Describes the way that machines providing published resources discover
Broker Desktops	Describes desktop concepts and usage.
Broker EntitlementPolicy	Controls end-user entitlement to desktop and application sessions provided
Broker ErrorHandling	Describes broker errors generated by cmdlets and how to access them.
Broker Filtering	Describes the common filtering options for XenDesktop cmdlets.
Broker Licensing	Overview of broker licensing configuration.
Broker Machines	Describes machine concepts and usage.
Broker Policies	Overview of the site policies that control users' access to desktop and
Broker PostInstallPreConfiguration	Describes how to configure the Citrix Broker Service port numbers, URL
Broker PowerManagement	Describes power management of machines used for desktops and applications.
Broker RemotePC	Overview of the Remote PC feature.

Cmdlets

Name	Description
Add-BrokerApplication	Adds applications to a desktop group.
Add-BrokerDesktopGroup	Associate Remote PC desktop groups with the specified Remote PC catalog.
Add-BrokerMachine	Adds one or more machines to a desktop group.
Add-BrokerMachineConfiguration	Adds a machine configuration to a desktop group.
Add-BrokerMachinesToDesktopGroup	Adds machines from a catalog to a desktop group.
Add-BrokerScope	Add the specified catalog/desktop group to the given scope(s).
Add-BrokerTag	Associate a tag with another object.
Add-BrokerUser	Creates an association between a user and another broker object
Disconnect-BrokerSession	Disconnect a session.
Export-BrokerDesktopPolicy	Gets the site wide Citrix Group Policy settings.
Get-BrokerAccessPolicyRule	Gets rules from the site's access policy.
Get-BrokerAppAssignmentPolicyRule	Gets application rules from the site's assignment policy.
Get-BrokerAppEntitlementPolicyRule	Gets application rules from the site's entitlement policy.
Get-BrokerApplication	Get the applications published on this site.
Get-BrokerApplicationInstance	Gets the running applications on the desktops.
Get-BrokerAssignmentPolicyRule	Gets desktop rules from the site's assignment policy.
Get-BrokerCatalog	Gets catalogs configured for this site.
Get-BrokerConfigurationSlot	Gets configuration slots configured for this site.

Name	Description
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 Citrix Broker Service's database connection string.
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 Citrix Broker Service database schema version information.
Get-BrokerMachine	Gets machines belonging to this site.
Get-BrokerMachineCommand	Get the list of commands queued for delivery to a desktop.

Name	Description
Get-BrokerMachineConfiguration	Gets machine configurations defined for this site.
Get-BrokerMachineStartMenuShortcutIcon	Retrieves a Start Menu Shortcut icon from the specified machine.
Get-BrokerMachineStartMenuShortcuts	Retrieves the Start Menu Shortcuts from the specified machine.
Get-BrokerPowerTimeScheme	Gets power management time schemes for desktop groups.
Get-BrokerPrivateDesktop	Get private desktops configured for this site.
Get-BrokerRebootCycle	Gets one or more reboot cycles.
Get-BrokerRebootSchedule	Gets one or more reboot schedules.
Get-BrokerRemotePCAccount	Get RemotePCAccount entries configured for this site.
Get-BrokerResource	Gets resources that a user can broker connections to.
Get-BrokerScopedObject	Gets the details of the scoped objects for the Broker Service.
Get-BrokerServiceAddedCapability	Gets any added capabilities for the Broker Service on the controller.
Get-BrokerServiceInstance	Gets the service instance entries for the Broker Service.
Get-BrokerServiceStatus	Determines the current state of the Broker Service on the controller.
Get-BrokerSession	Gets a list of sessions.
Get-BrokerSharedDesktop	Get shared desktops configured for this site.
Get-BrokerSite	Gets the current XenDesktop broker site.
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 Name	Description
	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.
Import-BrokerDesktopPolicy	Sets the site wide Citrix Group Policy settings for the site.
New-BrokerAccessPolicyRule	Creates a new rule in the site's access policy.
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.
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.

Name	Description
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-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-BrokerAppAssignmentPolicyRule	Deletes an application rule from the site's assignment policy.
Remove-BrokerAppEntitlementPolicyRule	Deletes an application rule from the site's entitlement policy.
Remove-BrokerApplication	Deletes one or more applications, or an association of an application.
Remove-BrokerApplicationInstanceMetadata	Deletes ApplicationInstance Metadata from the ApplicationInstance objects
Remove-BrokerApplicationMetadata	Deletes Application Metadata from the Application objects
Remove-BrokerAssignmentPolicyRule	Deletes a desktop rule from the site's assignment policy.
Remove-BrokerAssignmentPolicyRuleMetadata	Deletes AssignmentPolicyRule Metadata from the AssignmentPolicyRule objects
Remove-BrokerCatalog	Removes catalogs from the site.
Remove-BrokerCatalogMetadata	Deletes Catalog Metadata from the Catalog objects
Remove-BrokerConfigurationSlot	Removes a configuration slot.
Remove-BrokerConfigurationSlotMetadata	Deletes ConfigurationSlot Metadata from the ConfigurationSlot

Name	objects Description
Remove-BrokerConfiguredFTA	Deletes one or more configured file type associations.
Remove-BrokerControllerMetadata	Deletes Controller Metadata from the Controller objects
Remove-BrokerDelayedHostingPowerAction	Cancels one or more delayed power actions.
Remove-BrokerDesktopGroup	Remove desktop groups from the system or remove them from a Remote PC catalog.
Remove-BrokerDesktopGroupMetadata	Deletes DesktopGroup Metadata from the DesktopGroup objects
Remove-BrokerEntitlementPolicyRule	Deletes a desktop rule from the site's entitlement policy.
Remove-BrokerEntitlementPolicyRuleMetadata	Deletes EntitlementPolicyRule Metadata from the EntitlementPolicyRule objects
Remove-BrokerHostingPowerAction	Cancel one or more pending power actions.
Remove-BrokerHostingPowerActionMetadata	Deletes HostingPowerAction Metadata from the HostingPowerAction objects
Remove-BrokerHypervisorAlertMetadata	Deletes HypervisorAlert Metadata from the HypervisorAlert objects
Remove-BrokerHypervisorConnection	Removes a hypervisor connection from the system.
Remove-BrokerHypervisorConnectionMetadata	Deletes HypervisorConnection Metadata from the HypervisorConnection objects
Remove-BrokerIcon	Remove an icon.
Remove-BrokerIconMetadata	Deletes Icon Metadata from the Icon objects
Remove-BrokerImportedFTA	Deletes one or more imported file type associations.
Remove-BrokerMachine	Removes one or more machines from its desktop group or catalog.
Remove-BrokerMachineCommand	Cancel a pending command queued for delivery to a desktop.
Remove-BrokerMachineCommandMetadata	Deletes MachineCommand Metadata from the MachineCommand objects

Name	Description
Remove-BrokerMachineConfiguration	Deletes a machine configuration from the site or removes the association from a desktop group.
Remove-BrokerMachineConfigurationMetadata	Deletes MachineConfiguration Metadata from the MachineConfiguration objects
Remove-BrokerMachineMetadata	Deletes Machine Metadata from the Machine objects
Remove-BrokerPowerTimeScheme	Deletes an existing power time scheme.
Remove-BrokerPowerTimeSchemeMetadata	Deletes PowerTimeScheme Metadata from the PowerTimeScheme objects
Remove-BrokerRebootCycleMetadata	Deletes RebootCycle Metadata from the RebootCycle objects
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-BrokerSessionMetadata	Deletes Session Metadata from the Session objects
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-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.

Name	Description
Rename-BrokerCatalog	Renames a catalog.
Rename-BrokerDesktopGroup	Renames a desktop group.
Rename-BrokerEntitlementPolicyRule	Renames a desktop rule in the site's entitlement policy.
Rename-BrokerMachineConfiguration	Renames a machine configuration.
Rename-BrokerPowerTimeScheme	Changes the name of an existing power time scheme.
Rename-BrokerTag	Rename one or more tags.
Reset-BrokerLicensingConnection	Resets the broker's license server connection.
Reset-BrokerServiceGroupMembership	Makes the broker load, from the specified Configuration Service instance, the addresses that the Service can be reached on.
Send-BrokerSessionMessage	Sends a message to a session.
Set-BrokerAccessPolicyRule	Modifies an existing rule in the site's access policy.
Set-BrokerAccessPolicyRuleMetadata	Creates/Updates metadata key-value pairs for AccessPolicyRule
Set-BrokerAppAssignmentPolicyRule	Modifies an existing application rule in the site's assignment policy.
Set-BrokerAppEntitlementPolicyRule	Modifies an existing application rule in the site's entitlement policy.
Set-BrokerApplication	Changes the settings of an application to the value specified in the command.
Set-BrokerApplicationInstanceMetadata	Creates/Updates metadata key-value pairs for ApplicationInstance
Set-BrokerApplicationMetadata	Creates/Updates metadata key-value pairs for Application
Set-BrokerAssignmentPolicyRule	Modifies an existing desktop rule in the site's assignment policy.
Set-BrokerAssignmentPolicyRuleMetadata	Creates/Updates metadata key-value pairs for AssignmentPolicyRule
Set-BrokerCatalog	Sets the properties of a catalog.

Name	Description
Set-BrokerCatalogMetadata	Creates/Updates metadata key-value pairs for Catalog
Set-BrokerConfigurationSlotMetadata	Creates/Updates metadata key-value pairs for ConfigurationSlot
Set-BrokerControllerMetadata	Creates/Updates metadata key-value pairs for Controller
Set-BrokerDBConnection	Specifies the Citrix Broker Service's database connection string.
Set-BrokerDesktopGroup	Adjusts the settings of a broker desktop group.
Set-BrokerDesktopGroupMetadata	Creates/Updates metadata key-value pairs for DesktopGroup
Set-BrokerEntitlementPolicyRule	Modifies an existing desktop rule in the site's entitlement policy.
Set-BrokerEntitlementPolicyRuleMetadata	Creates/Updates metadata key-value pairs for EntitlementPolicyRule
Set-BrokerHostingPowerAction	Changes the priority of one or more pending power actions.
Set-BrokerHostingPowerActionMetadata	Creates/Updates metadata key-value pairs for HostingPowerAction
Set-BrokerHypervisorAlertMetadata	Creates/Updates metadata key-value pairs for HypervisorAlert
Set-BrokerHypervisorConnection	Sets the properties of a hypervisor connection.
Set-BrokerHypervisorConnectionMetadata	Creates/Updates metadata key-value pairs for HypervisorConnection
Set-BrokerIconMetadata	Creates/Updates metadata key-value pairs for Icon
Set-BrokerMachine	Sets properties on a machine.
Set-BrokerMachineCatalog	Moves one or more machines into a different catalog.
Set-BrokerMachineCommandMetadata	Creates/Updates metadata key-value pairs for MachineCommand
Set-BrokerMachineConfiguration	Sets the properties of a machine configuration.
Set-BrokerMachineConfigurationMetadata	Creates/Updates metadata key-value pairs for MachineConfiguration

Name	Description
Set-BrokerMachineMaintenanceMode	Specifies whether the specified machine(s) are in maintenance mode.
Set-BrokerMachineMetadata	Creates/Updates metadata key-value pairs for Machine
Set-BrokerPowerTimeScheme	Modifies an existing power time scheme.
Set-BrokerPowerTimeSchemeMetadata	Creates/Updates metadata key-value pairs for PowerTimeScheme
Set-BrokerPrivateDesktop	Change the settings of a private desktop.
Set-BrokerRebootCycleMetadata	Creates/Updates metadata key-value pairs for RebootCycle
Set-BrokerRebootSchedule	Updates the values of one or more desktop group reboot schedules.
Set-BrokerRemotePCAccount	Modify one or more RemotePCAccounts.
Set-BrokerSession	Sets properties of a session.
Set-BrokerSessionMetadata	Creates/Updates metadata key-value pairs for Session
Set-BrokerSharedDesktop	Change the settings of a shared desktop.
Set-BrokerSite	Changes the overall settings of the current XenDesktop broker site.
Set-BrokerSiteMetadata	Creates/Updates metadata key-value pairs for Site
Set-BrokerTagMetadata	Creates/Updates metadata key-value pairs for Tag
Start-BrokerCatalogPvdImagePrepare	Start the PVD Image prepare process in the Broker for the machines in the specified catalog(s).
Start-BrokerMachinePvdImagePrepare	Start the PVD Image prepare process in the Broker for the specified machine(s).
Start-BrokerNaturalRebootCycle	Reboots all machines from the specified catalog when they are not in use.
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.

Name	Description
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-BrokerNameCache	Performs administrative operations on the user and machine name cache.

about_Broker_AccessPolicy

Sep 29, 2015

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

Sep 29, 2015

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 BrowserName, PublishedName or the Name: \ / ; : # . * ? = < > | [] () " ' ,

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 "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 "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 "ACME\worker1"
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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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-BrokerMachineToDesktopGroup](#)

[Remove-BrokerMachine](#)

[Add-BrokerUser](#)

[Set-BrokerMachine](#)

[Set-BrokerPrivateDesktop](#)

[Get-BrokerMachine](#)

[Group-BrokerMachine](#)

[Get-BrokerDesktopUsage](#)

about_Broker_EntitlementPolicy

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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-BrokerRebootCycle
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

Sep 29, 2015

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

Sep 29, 2015
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](#)

[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

Sep 29, 2015
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

Sep 29, 2015
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 procees 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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

Creates an association between a user and another broker object

Syntax

```
Add-BrokerUser [-InputObject] <User[]> [-Application <Application>] [-Machine <Machine>] [-PrivateDesktop  
<PrivateDesktop>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Add-BrokerUser [-Name] <String> [-Application <Application>] [-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)

-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.

Disconnect-BrokerSession

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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, and Any.

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

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-BrokerAppAssignmentPolicyRule

Sep 29, 2015

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

Sep 29, 2015

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>] [-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).

-- 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

-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

Sep 29, 2015

Get the applications published on this site.

Syntax

```
Get-BrokerApplication [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerApplication [-Name] <String> [-ApplicationType <ApplicationType>] [-AssociatedDesktopGroupPriority <Int32>] [-AssociatedDesktopGroupUid <Int32>] [-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:

-- 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.

-- 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.

-- 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](#)

[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.

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

-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

-ClientFolder<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

-ShortcutAddedToStartMenu<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

-StartMenuFolder<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

Sep 29, 2015

Gets the running applications on the desktops.

Syntax

```
Get-BrokerApplicationInstance -Uid <Int64> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerApplicationInstance [-ApplicationName <String>] [-ApplicationUid <Int32>] [-Instances <Int32>] [-MachineName <String>] [-MachineUid <Int32>] [-Metadata <String>] [-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.

-- 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.

-- 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

-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

-SessionUid<Int64>

Gets only application instances for the published applications running in the specified session. Note, you must specify the SessionUid, not the SessionID.

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

Sep 29, 2015

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

Sep 29, 2015

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>] [-ProvisioningSchemeld <Guid>] [-ProvisioningType <ProvisioningType>] [-PvsAddress <String>] [-PvsDomain <String>] [-RemotePCDesktopGroupPriority <Int32>] [-RemotePCDesktopGroupUid <Int32>] [-RemotePCHypervisorConnectionUid <Int32>] [-Scopeld <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 the user changes are persisted on the machines in the catalog. Possible values are:

o OnLocal: The user changes are stored on the machine's local storage. o Discard: The user changes are discarded. o OnPvd: The 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

Required?	false
Default Value	
Accept Pipeline Input?	false

-PersistUserChanges<PersistUserChanges>

Specifies the location where the user changes will be persisted. Can only be set for single session catalogs. Values can be:

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the personal vDisk.

Required?	false
Default Value	
Accept Pipeline Input?	false

-ProvisioningSchemeId<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?	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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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>] [-LastStartTime <DateTime>] [-Metadata <String>] [-OSType <String>] [-OSVersion  
<String>] [-SID <String>] [-State <ControllerState>] [-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 instance of a controller running instances of the broker service. It contains the following properties:

-- ActiveSiteServices (System.String[])

The services active on the site

-- AssociatedHypervisorConnectionUids (System.Int32[])

The UIDs of the associated hypervisor connections

-- ControllerVersion (System.String)

The version of the controller

-- DesktopsRegistered (System.Int32)

The count of the desktops registered on the controller

-- DNSName (System.String)

The DNS name of the controller

-- LastActivityTime (System.DateTime?)

The last reported as active time of the controller

-- LastStartTime (System.DateTime?)

The last start-up time of the controller

-- MachineName (System.String)

The windows name of the controller

-- MetadataMap (System.Collections.Generic.Dictionary<string, string>)

The metadata for this command.

-- 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 controller.

-- Uid (System.Int32)

The UID of the controller itself.

Related topics

[Get-BrokerDesktop](#)

Parameters

-Uid<Int32>

Gets only Controllers 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 property 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 property 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 property is mainly of use with advanced filtering; see about_Broker_Filtering.

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

-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

Sep 29, 2015

Gets the Citrix Broker Service's database connection string.

Syntax

```
Get-BrokerDBConnection [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Gets the database connection string from the currently selected Citrix 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 that on the local machine, or that explicitly specified by the last usage of the -AdminAddress parameter to 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 in use by the currently selected Citrix Broker Service instance.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerDBConnection -AdminAddress controller1.mydomain.net
```

Gets the database connection string in use by the Citrix Broker Service instance running on the controller with FQDN "controller1.mydomain.net".

Get-BrokerDBSchema

Sep 29, 2015

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 that on the local machine, or that explicitly specified by the last usage of the -AdminAddress parameter to 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 what 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

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>

The name of the service group to be used when creating the Citrix Broker Service database schema. The service group is the collection of all broker services that share the same database and are considered equal (i.e. any service in the same service group can be used interchangeably).

Required?	false
Default Value	
Accept Pipeline Input?	false

-ScriptType<DatabaseScriptType>

Specifies the type of database script to be returned. The different script types are as follows:

-- 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>

Indicates that the database server is running on the same machine as the service instance itself. 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.

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

Sep 29, 2015

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.PSCustomObject

The Get-BrokerDBVersionChangeScript cmdlet returns a PSCustomObject 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.

Examples

----- EXAMPLE 1 -----

```
C:\PS> $update = Get-BrokerDBVersionChangeScript -DatabaseName MySiteDB -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

Sep 29, 2015

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

Sep 29, 2015

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>] [-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>] [-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[])

Username 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 was started without the use of the broker.

Session properties are always null for multi-session desktops.

-- 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.

-- 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 Virtual Delivery Agent running on the client connected.

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" or "RDP".

Session properties are always null for multi-session desktop.

-- 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, Agent Error.

-- 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 with the autonomously brokered session flag.

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.

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', or 'RDP'.

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 and Working.

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

Sep 29, 2015

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>] [-ScopeId <Guid>] [-ScopeName <String>] [-SecureIcaRequired <Boolean>] [-SessionSupport <SessionSupport>] [-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.

-- 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-BrokerAdministrator

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

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?	

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
------------------------	-------

-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

-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?	

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

Sep 29, 2015
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

Sep 29, 2015

Gets desktop rules from the site's entitlement policy.

Syntax

```
Get-BrokerEntitlementPolicyRule [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerEntitlementPolicyRule [[-Name] <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>] [-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:

-- 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.

-- 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

-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

-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?	

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

Sep 29, 2015

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

Sep 29, 2015

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?	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

Sep 29, 2015

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>] [-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.

-- 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

-HypHypervisorConnectionUid<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

-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

Sep 29, 2015
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

Sep 29, 2015

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 TXT FILE.

-- 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

Sep 29, 2015

Gets Citrix Broker Service database schema version information.

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>

Lists schema version numbers to which the current service schema could be upgraded.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Downgrade<SwitchParameter>

Lists schema version numbers to which the current service schema could be downgraded.

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.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerInstalledDBVersion
```

Gets the current Citrix Broker Service database schema version number.

----- **EXAMPLE 2** -----

```
C:\PS> Get-BrokerInstalledDBVersion -Upgrade
```

Gets the Citrix Broker Service database schema version numbers to which the service schema could be upgraded.

Get-BrokerMachine

Sep 29, 2015

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>] [-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>] [-LoadIndex <Int32>] [-MachineInternalState <MachineInternalState>] [-Metadata <String>] [-OSType <String>] [-OSVersion <String>] [-PersistUserChanges <PersistUserChanges>] [-PowerActionPending <Boolean>] [-PowerState <PowerState>] [-ProvisioningType <ProvisioningType>] [-PublishedApplication <String>] [-PublishedName <String>] [-PvdStage <PvdStage>] [-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>] [-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.

-- 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.

-- 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.

-- PvdStage (Citrix.Broker.Admin.SDK.PvdStage)

For a machine 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 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 was started without the use of the broker.

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 host name of the client connected to the machine.

Session properties are always null for multi-session machine.

-- 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 parameter that indicates if a session is hidden.

Session parameters 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" or "RDP".

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.

-- 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 sessions' 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 sessions' 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

-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

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

-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

-PvdStage<PvdStage>

Gets machines at a specific personal vDisk stage.

Valid values are None, Requested, PoweringOn and Working.

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 with the autonomously brokered session flag.

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?	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', or 'RDP'.

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

-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

Sep 29, 2015

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

Sep 29, 2015

Gets machine configurations defined for this site.

Syntax

```
Get-BrokerMachineConfiguration [-Uid] <Int32> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerMachineConfiguration [[-Name] <String>] [-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

-Leaf Name<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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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 -DesktopGroupUid ( Get-BrokerDesktopGroup 'Sales Desktops' ).Uid
```

Fetches all the power time schemes for the desktop group called 'Sales Desktops'.

Get-BrokerPrivateDesktop

Sep 29, 2015

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>] [-DesktopGroupId <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

-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

-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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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?	

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

PartialData

Only a subset of the available data was returned.

InvalidFilter

A filtering expression was included that could not be interpreted for this command.

PermissionDenied

You do not have permission to execute this command.

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 Catalog
```

```
ScopeId    : eff6f464-f1ee-4442-add3-99982e0cec01
ScopeName  : Sales
ObjectType : Catalog
ObjectId   : 1
ObjectName : MyExampleCatalog
Description : Test Catalog
```

```
ScopeId    : 304e0fa7-d390-47f0-a94f-7e956a324c41
ScopeName  : Finance
ObjectType : Catalog
ObjectId   : 1
ObjectName : MyExampleCatalog
Description : Test Catalog
```

```
ScopeId    :
ScopeName  :
ObjectType : Catalog
ObjectId   : 2
ObjectName : AnotherCatalog
Description : Another catalog in no scopes
```

Gets all of the scoped objects with type Catalog. The example output shows a catalog object (MyExampleCatalog) in two scopes Sales and Finance, and another catalog (AnotherCatalog) that is not in any scope. The ScopeId and ScopeName values returned are null in the final record.

Get-BrokerServiceAddedCapability

Sep 29, 2015

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.Collections.Generic.List<System.String>

List containing added capabilities.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerServiceAddedCapability
```

Gets the added capabilities of the Broker Service.

Get-BrokerServiceInstance

Sep 29, 2015

Gets the service instance entries for the Broker Service.

Syntax

```
Get-BrokerServiceInstance [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

This returns the service interfaces published by instances of the Broker Service. These instances can be used to register the Service with a Configuration Service so that other services are able to utilize their functionality. There is no requirement to have configured a database connection for the Service 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.Admin.SDK.ServiceInstance

Get-BrokerServiceInstance returns an opaque object containing information about the Broker WCF Service.

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-BrokerServiceInstance
```

Gets all the service instances of the Broker service running on the connected server. (If AdminAddress has not been set for the runspace, this is the Service running on the local machine.) To get the service instances of remote services, use the AdminAddress parameter. This defines the service required by the interfaces.

Get-BrokerServiceStatus

Sep 29, 2015

Determines the current state of the Broker Service on the controller.

Syntax

```
Get-BrokerServiceStatus [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

This command provides the ability for the status of the Broker Service in the controller to be determined. Before using this command, you don't have to configure the database connection to the Service.

Related topics

[Get-BrokerServiceInstance](#)

[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

Input Type

None You cannot pipe input into this cmdlet.

Return Values

System.String

Get-BrokerServiceStatus returns a string describing the status of the Broker Service.

DBUnconfigured

The broker does not have a database connection configured.

DBRejectedConnection

The database rejected the logon from the Broker Service. This may be caused by bad credentials, or the database not being installed.

InvalidDBConfigured

The database schema is missing (possibly just the stored procedures in it).

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 broker is too old to use the database. A newer version is required.

DBOlderVersionThanService

The database is too old for the Broker Service. Upgrade the database.

DBVersionChangeInProgress

A database schema upgrade is in progress.

OK

The broker is connected to a database that is valid, and the service is running.

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's status cannot be determined.

Examples

----- EXAMPLE 1 -----

```
C:\PS> Get-BrokerServiceStatus
```

Get the current status of the Service.

Get-BrokerSession

Sep 29, 2015

Gets a list of sessions.

Syntax

```
Get-BrokerSession -Uid <Int64> [-Property <String[]>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Get-BrokerSession [-AgentVersion <String>] [-ApplicationInUse <String>] [-AutonomouslyBrokered <Boolean>] [-
BrokeringDuration <Int32>] [-BrokeringTime <DateTime>] [-BrokeringUserName <String>] [-BrokeringUserSID <String>] [-
CatalogName <String>] [-ClientAddress <String>] [-ClientName <String>] [-ClientVersion <String>] [-ConnectedViaHostName
<String>] [-ConnectedViaIP <String>] [-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>] [-IsPhysical <Boolean>] [-LaunchedViaHostName
<String>] [-LaunchedViaIP <String>] [-MachineName <String>] [-MachineSummaryState <DesktopSummaryState>] [-
MachineUid <Int32>] [-Metadata <String>] [-OSType <String>] [-PersistUserChanges <PersistUserChanges>] [-PowerState
<PowerState>] [-Protocol <String>] [-ProvisioningType <ProvisioningType>] [-SecureIcaActive <Boolean>] [-SessionId
<Int32>] [-SessionKey <Guid>] [-SessionState <SessionState>] [-SessionStateChangeTime <DateTime>] [-SessionSupport
<SessionSupport>] [-SessionType <SessionType>] [-StartTime <DateTime>] [-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 SDK Name property of applications in use in the session.

-- AutonomouslyBrokered (System.Boolean)

Session property indicating if the session was started without the use of the broker.

-- BrokeringDuration (System.Int32?)

Time taken to broker the session.

-- 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 desktop.

-- ClientName (System.String)

The host name 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.

-- 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.

-- 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.

-- 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 either "HDX" or "RDP".

-- 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)

-- 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 PreparingNewSession, Connected, Active, Disconnected, Reconnecting, NonBrokeredSession, Other, and Unknown.

-- 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.

-- 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

-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

-AutonomouslyBrokered<Boolean>

Get autonomously brokered sessions.

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

-ClientName<String>

Get sessions by client name.

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

-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

-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

-MachineName<String>

Gets sessions by their machine name (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 and RDP.

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

-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

-SessionKey<Guid>

Gets session having the specified unique key.

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

-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-BrokerSharedDesktop

Sep 29, 2015

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

Sep 29, 2015

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.

-- 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

-- 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

Sep 29, 2015

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

Sep 29, 2015

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

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

Sep 29, 2015
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>] [-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
------------------------	-------

-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?	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

Sep 29, 2015

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 with the autonomously brokered session flag.

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

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', or 'RDP'.

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 and Working.

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

-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.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

Sep 29, 2015

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>] [-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>] [-LoadIndex <Int32>] [-MachineInternalState <MachineInternalState>] [-Metadata <String>] [-OSType <String>] [-OSVersion <String>] [-PersistUserChanges <PersistUserChanges>] [-PowerActionPending <Boolean>] [-PowerState <PowerState>] [-ProvisioningType <ProvisioningType>] [-PublishedApplication <String>] [-PublishedName <String>] [-PvdStage <PvdStage>] [-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>] [-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

-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

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

-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

-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

-PvdStage<PvdStage>

Gets machines at a specific personal vDisk stage.

Valid values are None, Requested, PoweringOn and Working.

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 with the autonomously brokered session flag.

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', or 'RDP'.

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

-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

-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 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.
```

Import-BrokerDesktopPolicy

Sep 29, 2015

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.

New-BrokerAccessPolicyRule

Sep 29, 2015
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, or allow access to any authenticated user. For a detailed description of this property see "help about_Broker_AccessPolicy".

Valid values are Filtered, AnyAuthenticated, and Any.

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-BrokerAppAssignmentPolicyRule

Sep 29, 2015

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

Sep 29, 2015

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[]>] [-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)
------------------------	-----------------------

-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

Sep 29, 2015
Creates a new published application.

Syntax

New-BrokerApplication [-Name] <String> -CommandLineExecutable <String> -DesktopGroup <DesktopGroup> [-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. 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 either the Name, BrowserName or the PublishedName: \ / : # . * ? = < > | [] () " ' .

See about_Broker_Applications for more information.

Related topics

- [Add-BrokerApplication](#)
- [Remove-BrokerApplication](#)
- [Get-BrokerApplication](#)
- [Remove-BrokerApplication](#)
- [Rename-BrokerApplication](#)
- [Set-BrokerApplication](#)

Parameters

-Name<String>

Specifies the unique name of the application.

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)

-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)

-ClientFolder<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)

-ShortcutAddedToDesktop<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 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

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

Sep 29, 2015
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

Sep 29, 2015
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> [-ProvisioningSchemeld <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 the location where the user changes will be persisted. Can only be set for single session catalogs. Values can be:

- o OnLocal - User changes are persisted locally.
- o Discard - User changes are discarded.
- o OnPvd - User changes are persisted on the 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

Required?	false
Default Value	The FunctionalLevel of the current release (L7); 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

Sep 29, 2015

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

Sep 29, 2015
Creates a file type association with a published application.

Syntax

New-BrokerConfiguredFTA -ImportedFTA <ImportedFTA> -ApplicationUid <Int32> [-UUID <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

New-BrokerConfiguredFTA -ExtensionName <String> -HandlerName <String> -ApplicationUid <Int32> [-ContentType <String>] [-HandlerDescription <String>] [-HandlerOpenArguments <String>] [-UUID <Guid>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

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

Sep 29, 2015

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

Sep 29, 2015

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>] [-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

Required?	false
Default Value	The FunctionalLevel of the current release (L7); 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)
------------------------	-----------------------

-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

Sep 29, 2015
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>] [-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)

-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 SUPPROT\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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

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 -UserId 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

Sep 29, 2015
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<String>]

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

Sep 29, 2015
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

Sep 29, 2015
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

Sep 29, 2015
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-BrokerTag

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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> Remove-BrokerAccessPolicyRuleMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the AccessPolicyRule in the site

Remove-BrokerAppAssignmentPolicyRule

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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](#)

[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

Sep 29, 2015

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> Remove-BrokerApplicationInstanceMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the ApplicationInstance in the site

Remove-BrokerApplicationMetadata

Sep 29, 2015

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> Remove-BrokerApplicationMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Application in the site

Remove-BrokerAssignmentPolicyRule

Sep 29, 2015

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

Sep 29, 2015

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> Remove-BrokerAssignmentPolicyRuleMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the AssignmentPolicyRule in the site

Remove-BrokerCatalog

Sep 29, 2015
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

- Add-BrokerCatalog
- [Get-BrokerCatalog](#)
- [New-BrokerCatalog](#)
- [Rename-BrokerCatalog](#)
- [Set-BrokerCatalog](#)
- [Add-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

Sep 29, 2015

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> Remove-BrokerCatalogMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Catalog in the site

Remove-BrokerConfigurationSlot

Sep 29, 2015

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

Sep 29, 2015

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> Remove-BrokerConfigurationSlotMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the ConfigurationSlot in the site

Remove-BrokerConfiguredFTA

Sep 29, 2015

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

Sep 29, 2015

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> Remove-BrokerControllerMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Controller in the site

Remove-BrokerDelayedHostingPowerAction

Sep 29, 2015

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

Sep 29, 2015

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](#)

[Add-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

Sep 29, 2015

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> Remove-BrokerDesktopGroupMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the DesktopGroup in the site

Remove-BrokerEntitlementPolicyRule

Sep 29, 2015

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

Sep 29, 2015

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> Remove-BrokerEntitlementPolicyRuleMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the EntitlementPolicyRule in the site

Remove-BrokerHostingPowerAction

Sep 29, 2015

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

Sep 29, 2015

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> Remove-BrokerHostingPowerActionMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the HostingPowerAction in the site

Remove-BrokerHypervisorAlertMetadata

Sep 29, 2015

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> Remove-BrokerHypervisorAlertMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the HypervisorAlert in the site

Remove-BrokerHypervisorConnection

Sep 29, 2015

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

Sep 29, 2015

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> Remove-BrokerHypervisorConnectionMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the HypervisorConnection in the site

Remove-BrokerIcon

Sep 29, 2015
Remove an icon.

Syntax

Remove-BrokerIcon [-InputObject] <Icon[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Detailed Description

Removes an icon from the database.

Related topics

Add-BrokerIcon

[New-BrokerIcon](#)

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

Sep 29, 2015

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> Remove-BrokerIconMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Icon in the site

Remove-BrokerImportedFTA

Sep 29, 2015

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-BrokerMachine

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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> Remove-BrokerMachineCommandMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the MachineCommand in the site

Remove-BrokerMachineConfiguration

Sep 29, 2015

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

Sep 29, 2015

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> Remove-BrokerMachineConfigurationMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the MachineConfiguration in the site

Remove-BrokerMachineMetadata

Sep 29, 2015

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> Remove-BrokerMachineMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Machine in the site

Remove-BrokerPowerTimeScheme

Sep 29, 2015
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

Sep 29, 2015

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> Remove-BrokerPowerTimeSchemeMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the PowerTimeScheme in the site

Remove-BrokerRebootCycleMetadata

Sep 29, 2015

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> Remove-BrokerRebootCycleMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the RebootCycle in the site

Remove-BrokerRebootSchedule

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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-BrokerSessionMetadata

Sep 29, 2015

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> Remove-BrokerSessionMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Session in the site

Remove-BrokerSiteMetadata

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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> Remove-BrokerTagMetadata -Name "MyMetadataName"
```

This command deletes the Metadata "MyMetadataName" key-value pair for all the Tag in the site

Remove-BrokerUser

Sep 29, 2015

Remove broker user objects from another broker object

Syntax

```
Remove-BrokerUser [-InputObject] <User[]> [-Application <Application>] [-Machine <Machine>] [-PrivateDesktop  
<PrivateDesktop>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-BrokerUser [-Name] <String> [-Application <Application>] [-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)

-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

Sep 29, 2015
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-BrokerAppAssignmentPolicyRule

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015

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?	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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

Makes the broker load, from the specified Configuration Service instance, the addresses that the Service can be reached on.

Syntax

```
Reset-BrokerServiceGroupMembership -ConfigServiceInstance <PSObject[]> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

Detailed Description

This cmdlet reloads the access permissions and Configuration Service locations for the Broker Service. It must be run on at least one instance of the Broker Service after installation and registration with the Configuration Service. Without this operation, the Broker Services cannot communicate with the other services that form part of the Citrix XenDesktop deployment. Once the cmdlet is run, the services keep up to date when additional services are added to the deployment provided that the Configuration Service is not stopped. Run this cmdlet 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 cmdlet, only the first is used that meets the expected service type requirements.

Related topics

[Get-BrokerServiceInstance](#)

[Get-BrokerServiceStatus](#)

[Get-ConfigRegisteredServiceInstance](#)

Parameters

-ConfigServiceInstance<PSObject[]>

The Configuration Service Instance object that represents a service instance of the type InterService that in turn 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.XDPowerShell.ServiceInstance You can pipe a ServiceInstance that contains a ServiceInstance object that in turn refers to the Configuration Service InterService interface.

Return Values

Citrix.XDPowerShell.ServiceInstance

Reset-BrokerServiceGroupMembership returns opaque objects containing Configuration Service instances that are used by the Broker Service instance.

Examples

----- EXAMPLE 1 -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config | Reset-BrokerServiceGroupMembership
```

Resets the Broker Service's group membership. This assumes that the Configuration Service is running and configured on the same machine as the Broker service.

----- EXAMPLE 2 -----

```
c:\PS>Get-ConfigRegisteredServiceInstance -ServiceType Config -AdminAddress OtherServer.com | Reset-BrokerServiceGroupmembership
```

Resets the Broker Service's group membership. This uses a Configuration Service that is running and configured on a server called OtherServer.com.

Send-BrokerSessionMessage

Sep 29, 2015

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

Sep 29, 2015
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?	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, or allow access to any authenticated user. For a detailed description of this property see "help about_Broker_AccessPolicy".

Valid values are Filtered, AnyAuthenticated, and Any.

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

Sep 29, 2015

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-BrokerAppAssignmentPolicyRule

Sep 29, 2015

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

Sep 29, 2015

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[]>] [-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[]>] [-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

-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

Sep 29, 2015

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](#)

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

--	--

-ClientFolder<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

Sep 29, 2015

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> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationInstanceMetadata [-ApplicationInstanceId] <Int64> -Map <PSObject> [-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 (ByValue)

-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

Sep 29, 2015

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[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-InputObject] <Application[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-ApplicationName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerApplicationMetadata [-ApplicationName] <String> -Name <String> -Value <String> [-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

Sep 29, 2015

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?	

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

Sep 29, 2015

Creates/Updates metadata key-value pairs for AssignmentPolicyRule

Syntax

```
Set-BrokerAssignmentPolicyRuleMetadata [-AssignmentPolicyRuleId] <Int32> -Map <PObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
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 [-InputObject] <AssignmentPolicyRule[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-InputObject] <AssignmentPolicyRule[]> -Map <PObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-AssignmentPolicyRuleName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerAssignmentPolicyRuleMetadata [-AssignmentPolicyRuleName] <String> -Map <PObject> [-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, ByPropertyName)

-InputObject<AssignmentPolicyRule[]>

Specifies the AssignmentPolicyRule objects whose Metadata is to be created/updated.

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue)

-Assignment PolicyRuleName<String>

Specifies the AssignmentPolicyRule 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.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

Sep 29, 2015
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

Required?	false
Default Value	
Accept Pipeline Input?	false

-ProvisioningSchemeld<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

Sep 29, 2015

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[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-InputObject] <Catalog[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-CatalogName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [-<CommonParameters>]
```

```
Set-BrokerCatalogMetadata [-CatalogName] <String> -Name <String> -Value <String> [-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

Sep 29, 2015

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> -Map <PObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-ConfigurationSlotId] <Int32> -Name <String> -Value <String> [-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 <PObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-ConfigurationSlotName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerConfigurationSlotMetadata [-ConfigurationSlotName] <String> -Map <PObject> [-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

Sep 29, 2015

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> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerControllerMetadata [-ControllerName] <String> -Map <PSObject> [-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 (ByPropertyName)

-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

Sep 29, 2015

Specifies the Citrix Broker Service's database connection string.

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.

Is 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 that on the local machine, or that explicitly specified by the last usage of the -AdminAddress parameter to a Broker SDK cmdlet.

Related topics

[Get-BrokerServiceStatus](#)

[Get-BrokerDBConnection](#)

[Test-BrokerDBConnection](#)

Parameters

-DBConnection<String>

Specifies the new database connection string for the currently selected Citrix Broker Service instance.

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	
Accept Pipeline Input?	false

Input Type

None You cannot pipe input into this cmdlet.

Return Values

System.String

The Set-BrokerDBConnection cmdlet returns a string describing the new status of the selected Citrix Broker Service

instance. Possible values are:

-- OK:

The service instance is configured with a valid database and service schema. The service is operational.

-- DBUnconfigured:

No database connection string is set for the service instance.

-- DBRejectedConnection:

The database server rejected the logon from the service instance. This is typically caused by invalid credentials.

-- InvalidDBConfigured:

The specified database does not exist, is not visible to the 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 service instance is older than, and incompatible with, the service's schema in the database. The service instance needs upgrading.

-- DBOlderVersionThanService:

The 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 in progress.

-- PendingFailure:

Connectivity between the 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 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.

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

Sep 29, 2015

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>] [-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>] [-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

-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

Sep 29, 2015

Creates/Updates metadata key-value pairs for DesktopGroup

Syntax

```
Set-BrokerDesktopGroupMetadata [-DesktopGroupId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
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 [-InputObject] <DesktopGroup[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerDesktopGroupMetadata [-InputObject] <DesktopGroup[]> -Name <String> -Value <String> [-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 (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)

-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.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

Sep 29, 2015

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>] [-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>] [-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

-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

Sep 29, 2015

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[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerEntitlementPolicyRuleMetadata [-InputObject] <EntitlementPolicyRule[]> -Name <String> -Value <String> [-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 (ByPropertyName)

-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

Sep 29, 2015

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

Sep 29, 2015

Creates/Updates metadata key-value pairs for HostingPowerAction

Syntax

```
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 [-HostingPowerActionId] <Int64> -Map <PSObject> [-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 Input Object 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 (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)

-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.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

Sep 29, 2015

Creates/Updates metadata key-value pairs for HypervisorAlert

Syntax

```
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 [-HypervisorAlertId] <Int64> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorAlertMetadata [-InputObject] <HypervisorAlert[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorAlertMetadata [-InputObject] <HypervisorAlert[]> -Name <String> -Value <String> [-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 (ByValue, ByPropertyName)

-InputObject<HypervisorAlert[]>

Specifies the HypervisorAlert 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.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

Sep 29, 2015

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

Sep 29, 2015

Creates/Updates metadata key-value pairs for HypervisorConnection

Syntax

```
Set-BrokerHypervisorConnectionMetadata [-HypervisorConnectionId] <Int32> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
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 [-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> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerHypervisorConnectionMetadata [-HypervisorConnectionName] <String> -Map <PSObject> [-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 (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)

-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.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

Sep 29, 2015

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[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Set-BrokerIconMetadata [-InputObject] <Icon[]> -Name <String> -Value <String> [-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-BrokerMachine

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

Creates/Updates metadata key-value pairs for MachineCommand

Syntax

```
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 [-MachineCommandId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineCommandMetadata [-InputObject] <MachineCommand[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineCommandMetadata [-InputObject] <MachineCommand[]> -Map <PSObject> [-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)

-InputObject<MachineCommand[]>

Specifies the MachineCommand 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.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

Sep 29, 2015

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?	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

Sep 29, 2015

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[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-InputObject] <MachineConfiguration[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-MachineConfigurationName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineConfigurationMetadata [-MachineConfigurationName] <String> -Name <String> -Value <String> [-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

Sep 29, 2015

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
```

This example finds all machines in domain MyDomain and removes them from maintenance mode by setting their InMaintenanceMode property to false.

Set-BrokerMachineMetadata

Sep 29, 2015

Creates/Updates metadata key-value pairs for Machine

Syntax

```
Set-BrokerMachineMetadata [-MachineId] <Int32> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
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 [-InputObject] <Machine[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-InputObject] <Machine[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-MachineName] <String> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerMachineMetadata [-MachineName] <String> -Name <String> -Value <String> [-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 (ByValue, 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)

-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.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

Sep 29, 2015
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

Sep 29, 2015

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[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-InputObject] <PowerTimeScheme[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-PowerTimeSchemeName] <String> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerPowerTimeSchemeMetadata [-PowerTimeSchemeName] <String> -Map <PSObject> [-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 (ByValue)

-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

Sep 29, 2015

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

-AssignedClientName<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 PassThru 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

Sep 29, 2015

Creates/Updates metadata key-value pairs for RebootCycle

Syntax

```
Set-BrokerRebootCycleMetadata [-RebootCycleId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
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 [-InputObject] <RebootCycle[]> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerRebootCycleMetadata [-InputObject] <RebootCycle[]> -Name <String> -Value <String> [-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, ByPropertyName)

-InputObject<RebootCycle[]>

Specifies the RebootCycle 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.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

Sep 29, 2015
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

-InputObject<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

-RebootDuration<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

Sep 29, 2015

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

Sep 29, 2015

Sets properties of a session.

Syntax

```
Set-BrokerSession [-InputObject] <Session[]> [-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 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)

-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-BrokerSessionMetadata

Sep 29, 2015

Creates/Updates metadata key-value pairs for Session

Syntax

```
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 [-SessionId] <Int64> -Map <PSObject> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-InputObject] <Session[]> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerSessionMetadata [-InputObject] <Session[]> -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 (ByPropertyName)

-InputObject<Session[]>

Specifies the Session 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.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

Set-BrokerSharedDesktop

Sep 29, 2015
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

Sep 29, 2015

Changes the overall settings of the current XenDesktop broker site.

Syntax

```
Set-BrokerSite [-PassThru] [-BaseOU <Guid>] [-ColorDepth <ColorDepth>] [-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

-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

Sep 29, 2015

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

Sep 29, 2015

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> -Name <String> -Value <String> [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-BrokerTagMetadata [-TagName] <String> -Map <PSObject> [-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 (ByValue)

-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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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 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 Director typically create High Level Operations. PowerShell scripts can also wrap a series of cmdlet calls in a High Level Operation by way of 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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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 that on the local machine, or that explicitly specified by the last usage of the -AdminAddress parameter to 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

System.String

The Test-BrokerDBConnection cmdlet returns a string describing the status of the selected Citrix Broker Service instance that would result if the connection string were used with the Set-BrokerDBConnection cmdlet; the actual current status of the service is not changed. Possible values are:

-- OK:

The service instance is configured with a valid database and service schema. The service is operational.

-- DBUnconfigured:

No database connection string is set for the service instance.

-- DBRejectedConnection:

The database server rejected the logon from the service instance. This is typically caused by invalid credentials.

-- InvalidDBConfigured:

The specified database does not exist, is not visible to the 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 service instance is older than, and incompatible with, the service's schema in the database. The service instance needs upgrading.

-- DBOlderVersionThanService:

The 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 in progress.

-- PendingFailure:

Connectivity between the 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 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.

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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-BrokerNameCache

Sep 29, 2015

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

Sep 29, 2015
Overview

Name	Description
ConfigConfigurationSnapin	The Configuration service PowerShell snap-in provides administrative
Config Filtering	Describes the common filtering options for XenDesktop cmdlets.

Cmdlets

Name	Description
Add-ConfigRegisteredServiceInstanceMetadata	Adds metadata on the given ServiceInstance.
Add-ConfigServiceGroupMetadata	Adds metadata on the given ServiceGroup.
Export-ConfigFeatureTable	Returns the current feature table.
Get-ConfigDBConnection	Gets the database string for the specified data store used by the Configuration Service.
Get-ConfigDBSchema	Gets a script that creates the Configuration Service database schema for the specified data store.
Get-ConfigDBVersionChangeScript	Gets a script that updates the Configuration Service database schema.
Get-ConfigEnabledFeature	Lists features of the site that are enabled.
Get-ConfigInstalledDBVersion	Gets a list of all available database schema versions for the Configuration Service.
Get-ConfigLicensingModel	Lists the supported licensing models.
Get-ConfigLocalData	Gets the service local data.
Get-ConfigProduct	Lists the site's supported product names and codes.
Get-ConfigProductEdition	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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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 \/:#. *?=<> | []()''

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

Sep 29, 2015
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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015

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

-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 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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

PSObject

Examples

----- **EXAMPLE 1** -----

```
C:\PS> Get-ConfigProduct
```

Lists the supported products by name and code.

Get-ConfigProductEdition

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

Gets the service instances that are registered in the directory.

Syntax

```
Get-ConfigRegisteredServiceInstance [-ServiceInstanceId <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

-ServiceInstanceId<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_Configfiltering for details.

Required?	false
Default Value	false
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

See about_Configfiltering for details.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

See about_Configfiltering for details.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

See about_Configfiltering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

-Filter<String>

See about_Configfiltering 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\$
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\$
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\$
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\$
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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

Gets the service groups that match the parameters supplied.

Syntax

```
Get-ConfigServiceGroup [-ServiceGroupUid <Guid>] [-ServiceGroupName <String>] [-ServiceType <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
------------------------	-------

-ReturnTotalRecordCount<SwitchParameter>

See about_Configfiltering for details.

Required?	false
Default Value	false
Accept Pipeline Input?	false

-MaxRecordCount<Int32>

See about_Acctfiltering for details.

Required?	false
Default Value	250
Accept Pipeline Input?	false

-Skip<Int32>

See about_Acctfiltering for details.

Required?	false
Default Value	0
Accept Pipeline Input?	false

-SortBy<String>

See about_Acctfiltering for details.

Required?	false
Default Value	
Accept Pipeline Input?	false

--	--

-Filter<String>

See about_Acctfiltering 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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

Allows the registration of a service instance.

Syntax

```
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 -ServiceGroupUid <Guid> -ServiceGroupName <String> -ServiceType <String> -Address <String> -Binding <String> -Version <Int32> -ServiceAccountSid <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)

-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)

-InterfaceType<String>

The type of interface that the service provides (i.e. SDK, InterService, Peer).

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)

-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.

ServiceInstanceId <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

Sep 29, 2015

Removes metadata from the given ServiceInstance.

Syntax

```
Remove-ConfigRegisteredServiceInstanceMetadata [-ServiceInstanceId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ConfigRegisteredServiceInstanceMetadata [-ServiceInstanceId] <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

-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>

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015
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

Sep 29, 2015
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

Sep 29, 2015
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[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-ConfigRegisteredServiceInstanceMetadata [-InputObject] <ServiceInstance[]> -Name <String> -Value <String> [-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 Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of 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 -ServiceInstanceId 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

Sep 29, 2015
Adds or updates metadata on the given ServiceGroup.

Syntax

```
Set-ConfigServiceGroupMetadata [-ServiceGroupUid] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]  
  
Set-ConfigServiceGroupMetadata [-ServiceGroupUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]  
  
Set-ConfigServiceGroupMetadata [-InputObject] <ServiceGroup[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]  
  
Set-ConfigServiceGroupMetadata [-InputObject] <ServiceGroup[]> -Name <String> -Value <String> [-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)

-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

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

Sep 29, 2015

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[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-ConfigServiceMetadata [-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-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

Sep 29, 2015
Changes the overall settings of the site.

Syntax

```
Set-ConfigSite [-SiteName <String>] [-ProductCode <String>] [-ProductEdition <String>] [-ProductVersion <String>] [-LicensingModel <String>] [-LicenseServerName <String>] [-LicenseServerPort <Int32>] [-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

-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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

Overview

Name	Description
LogConfigurationLoggingSnapin	The Configuration Logging Service PowerShell snap-in provides administrative
Log Filtering	Describes the common filtering options for XenDesktop cmdlets.
logical operators	Describes the operators that connect statements in Windows PowerShell.

Cmdlets

Name	Description
Export-LogReportCsv	Exports Configuration Logging data into a CSV file.
Export-LogReportHtml	Exports Configuration Logging data into a HTML report.
Get-LogDataStore	Gets details for each of the ConfigurationLogging data stores.
Get-LogDBConnection	Gets the database string for the specified data store used by the ConfigurationLogging Service.
Get-LogDBSchema	Gets a script that creates the ConfigurationLogging Service database schema for the specified data store.
Get-LogDBVersionChangeScript	Gets a script that updates the ConfigurationLogging Service database schema.
Get-LogHighLevelOperation	Gets high level operations
Get-LogInstalledDBVersion	Gets a list of all available database schema versions for the ConfigurationLogging Service.
Get-LogLowLevelOperation	Gets low level operations
Get-LogService	Gets the service record entries for the ConfigurationLogging Service.
Get-LogServiceAddedCapability	Gets any added capabilities for the ConfigurationLogging Service on the controller.

Name	Description
Get-LogServiceInstance	Gets the service instance entries for the ConfigurationLogging Service.
Get-LogServiceStatus	Gets the current status of the ConfigurationLogging Service on the controller.
Get-LogSite	Gets global configuration logging settings.
Get-LogSummary	Gets operations logged within time intervals inside a date range.
Remove-LogOperation	Deletes configuration logs
Remove-LogServiceMetadata	Removes metadata from the given Service.
Remove-LogSiteMetadata	Removes metadata from the given Site.
Reset-LogDataStore	Refreshes the database string currently being used by the Log service.
Reset-LogServiceGroupMembership	Reloads the access permissions and configuration service locations for the ConfigurationLogging Service.
Set-LogDBConnection	Configures a database connection for the ConfigurationLogging Service.
Set-LogServiceMetadata	Adds or updates metadata on the given Service.
Set-LogSite	Sets global configuration logging settings.
Set-LogSiteMetadata	Adds or updates metadata on the given Site.
Start-LogHighLevelOperation	Logs the start of a high level operation.
Stop-LogHighLevelOperation	Logs the completion of a previously started high level operation.
Test-LogDBConnection	Tests a database connection for the ConfigurationLogging Service.

about_LogConfigurationLoggingSnapin

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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) False
-or	Logical or. TRUE when either or both statements are TRUE.	(1 -eq 1) -or (1 -eq 2) 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) False
-not	Logical not. Negates the statement that follows it.	-not (1 -eq 1) False
!	Logical not. Negates the statement that follows it. (Same as -not)	!(1 -eq 1) 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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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>] [-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

-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.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

Sep 29, 2015

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

Sep 29, 2015

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>] [-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

-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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

Deletes configuration logs

Syntax

```
Remove-LogOperation -DatabaseCredentials <PSCredential> [-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>]
```

```
Remove-LogOperation -UserName <String> [-Password <String>] [-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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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	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

Sep 29, 2015
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

Sep 29, 2015

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[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-LogServiceMetadata [-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-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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015
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

Sep 29, 2015
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

Sep 29, 2015

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	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

Sep 29, 2015

Overview

Name	Description
AdminDelegatedAdminSnapin	The Delegated Administration Service PowerShell snap-in provides
Admin Filtering	Describes the common filtering options for XenDesktop cmdlets.

Cmdlets

Name	Description
Add-AdminPermission	Add permissions to the set of permissions of a role.
Add-AdminRight	Grants a given right to the specified administrator.
Get-AdminAdministrator	Gets administrators configured for this site.
Get-AdminDBConnection	Gets the database string for the specified data store used by the DelegatedAdmin Service.
Get-AdminDBSchema	Gets a script that creates the DelegatedAdmin Service database schema for the specified data store.
Get-AdminDBVersionChangeScript	Gets a script that updates the DelegatedAdmin Service database schema.
Get-AdminEffectiveAdministrator	Retrieve the effective administrator objects for a user.
Get-AdminEffectiveRight	Gets the set of Right objects associated with the current user.
Get-AdminInstalledDBVersion	Gets a list of all available database schema versions for the DelegatedAdmin Service.
Get-AdminPermission	Gets permissions configured for the site.
Get-AdminPermissionGroup	Gets permission groups configured for the site.
Get-AdminRevision	Gets the current revision of the delegated administration configuration data.

Get-AdminRole Name	Gets roles configured for this site. 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 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

Sep 29, 2015

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 superset 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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015

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

-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 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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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 superset 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

Sep 29, 2015

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

Sep 29, 2015

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

-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.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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015
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?	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

Sep 29, 2015

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

Sep 29, 2015

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?	

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

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-HypHypervisorConnectionScope -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

Sep 29, 2015

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

Sep 29, 2015

Removes metadata from the given Administrator.

Syntax

```
Remove-AdminAdministratorMetadata -AdministratorSid <String> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminAdministratorMetadata -AdministratorSid <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminAdministratorMetadata [-AdministratorName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-AdminAdministratorMetadata [-AdministratorName] <String> -Name <String> [-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)

-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-AdminAdministrator | % { Remove-AdminAdministratorMetadata -Map $_.MetadataMap }  
Remove all metadata from all Administrator objects.
```


Remove-AdminPermission

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
Removes a role from the site.

Syntax

```
Remove-AdminRole [-InputObject] <Role[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]  
Remove-AdminRole [-Id] <Guid[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]  
Remove-AdminRole [-Name] <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-AdminRole cmdlet deletes roles from the site.

You cannot remove built-in roles or roles that are currently assigned to an administrator.

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)

-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

Sep 29, 2015

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[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminRoleMetadata [-InputObject] <Role[]> -Name <String> [-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

Sep 29, 2015
Removes a scope from the site.

Syntax

```
Remove-AdminScope [-InputObject] <Scope[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]  
Remove-AdminScope [-Id] <Guid[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]  
Remove-AdminScope [-Name] <String[]> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

The Remove-AdminScope cmdlet deletes scopes from the site.

You cannot remove the built-in 'All' scope or scopes that are currently assigned to an administrator.

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)

-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

Sep 29, 2015

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[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminScopeMetadata [-InputObject] <Scope[]> -Map <PSObject> [-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

Sep 29, 2015

Removes metadata from the given Service.

Syntax

```
Remove-AdminServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminServiceMetadata [-InputObject] <Service[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-AdminServiceMetadata [-InputObject] <Service[]> -Name <String> [-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)

-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-AdminService | % { Remove-AdminServiceMetadata -Map $_.MetadataMap }
```

Remove all metadata from all Service objects.

Rename-AdminRole

Sep 29, 2015
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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-AdminAdministratorMetadata [-InputObject] <Administrator[]> -Map <PSObject> [-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

Sep 29, 2015
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

Sep 29, 2015
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

Sep 29, 2015
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[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-AdminRoleMetadata [-InputObject] <Role[]> -Name <String> -Value <String> [-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

Sep 29, 2015

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

Sep 29, 2015
Adds or updates metadata on the given Scope.

Syntax

Set-AdminScopeMetadata [-ScopeId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-AdminScopeMetadata [-ScopeId] <Guid> -Map <PSObject> [-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[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-AdminScopeMetadata [-InputObject] <Scope[]> -Name <String> -Value <String> [-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)

-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

-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-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

Sep 29, 2015

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

Sep 29, 2015

Retrieves the scopes where the specified operation is permitted.

Syntax

```
Test-AdminAccess [-Operation] <String> [-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.

Related topics

Parameters

-Operation<String>

The operation 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

Return Values

Citrix.DelegatedAdmin.Sdk.ScopeReference

The list of permissible scopes for the specified operation.

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.

Test-AdminDBConnection

Sep 29, 2015
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

Sep 29, 2015

Overview

Name	Description
EnvTestEnvTestSnapin	The Citrix Environment Test Service provides tools to test and inspect the state of a XenDesktop installation.
EnvTest Filtering	Describes the common filtering options for XenDesktop cmdlets.

Cmdlets

Name	Description
Get-EnvTestConfiguration	Gets the Environment Test Service's configuration options
Get-EnvTestDBConnection	Gets the database string for the specified data store used by the EnvTest Service.
Get-EnvTestDBSchema	Gets a script that creates the EnvTest Service database schema for the specified data store.
Get-EnvTestDBVersionChangeScript	Gets a script that updates the EnvTest Service database schema.
Get-EnvTestDefinition	Gets the one or more test definitions
Get-EnvTestInstalledDBVersion	Gets a list of all available database schema versions for the EnvTest Service.
Get-EnvTestService	Gets the service record entries for the EnvTest Service.
Get-EnvTestServiceAddedCapability	Gets any added capabilities for the EnvTest Service on the controller.
Get-EnvTestServiceInstance	Gets the service instance entries for the EnvTest Service.
Get-EnvTestServiceStatus	Gets the current status of the EnvTest Service on the controller.
Get-EnvTestSuiteDefinition	Gets one or more test suite definitions.
Get-EnvTestTask	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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

-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 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

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-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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

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** -----

```
$taskOfSpecifidId = 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

Sep 29, 2015
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

Sep 29, 2015

Removes metadata from the given Service.

Syntax

```
Remove-EnvTestServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-EnvTestServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-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)

-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-EnvTestService | %{ Remove-EnvTestServiceMetadata -Map $_.MetadataMap }
```

Remove all metadata from all Service objects.

Remove-EnvTestTask

Sep 29, 2015
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, retrieveable 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

Sep 29, 2015

Removes metadata from the given Task.

Syntax

```
Remove-EnvTestTaskMetadata [-TaskTaskId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-EnvTestTaskMetadata [-TaskTaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-EnvTestTaskMetadata [-InputObject] <EnvTestTask[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Remove-EnvTestTaskMetadata [-InputObject] <EnvTestTask[]> -Map <PSObject> [-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)

-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-EnvTestTask | % { Remove-EnvTestTaskMetadata -Map $_.MetadataMap }  
Remove all metadata from all Task objects.
```

Reset-EnvTestServiceGroupMembership

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

Adds or updates metadata on the given Service.

Syntax

```
Set-EnvTestServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]
```

```
Set-EnvTestServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

```
Set-EnvTestServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]
```

```
Set-EnvTestServiceMetadata [-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-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)

-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 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

Sep 29, 2015

Adds or updates metadata on the given Task.

Syntax

```
Set-EnvTestTaskMetadata [-TaskTaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-EnvTestTaskMetadata [-TaskTaskId] <Guid> -Name <String> -Value <String> [-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)

-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?	

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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015
Overview

Name	Description
HypHostSnapin	The Host Service PowerShell snap-in provides administrative functions for
Hyp Filtering	Describes the common filtering options for XenDesktop cmdlets.

Cmdlets

Name	Description
Add-HypHostingUnitMetadata	Adds metadata on the given HostingUnit.
Add-HypHostingUnitNetwork	Makes additional hypervisor networks available for use in a HostingUnit.
Add-HypHostingUnitStorage	Adds storage locations to a hosting unit.
Add-HypHypervisorConnectionAddress	Add a connection address to a hypervisor connection.
Add-HypHypervisorConnectionMetadata	Adds metadata on the given HypervisorConnection.
Add-HypHypervisorConnectionScope	Add the specified HypervisorConnection(s) to the given scope(s).
Add-HypMetadata	Adds metadata to a hypervisor connection or a hosting unit.
Get-HypConfigurationDataForItem	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.
Get-HypConfigurationObjectForItem	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.
Get-HypConnectionRegion	Enumerates the regions of a hypervisor connection that are based on cloud technology.
Get-HypDBConnection	Gets the database string for the specified data store used by the Host Service.

Get-HypDBSchema Name	Description
	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- Name	Description
Remove-HypHypervisorConnectionMetadata	Removes metadata from the given HypervisorConnection.
Remove-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

HypervisorConnectionNameAvailable Name	unused. Description
Update-HypervisorConnection	Requests the host service to update the connection properties that depend on the version of hypervisor in use.

about_HypHostSnapin

Sep 29, 2015

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 runspace 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
-Passwd 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
PSProvider:     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\<Name>

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\<Name>
XDHyp:\Connections\{Guid}
XDHyp:\HostingUnits\<Name>
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\<Name>
XDHyp:\Connections\{Guid}
XDHyp:\HostingUnits\<Name>
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

Sep 29, 2015

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

Sep 29, 2015
Adds metadata on the given HostingUnit.

Syntax

Add-HypHostingUnitMetadata [-HostingUnitUid] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Add-HypHostingUnitMetadata [-HostingUnitUid] <Guid> -Map <PSObject> [-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[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Add-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. 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)

-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

-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

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

Sep 29, 2015
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

Sep 29, 2015
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.

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 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

Sep 29, 2015
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

Sep 29, 2015
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[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

Add-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. 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 Director typically create high-level operations. PowerShell scripts can also wrap a series of cmdlet calls in a high-level operation by way of 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-HypHypervisorConnectionMetadata 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

Sep 29, 2015
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

Sep 29, 2015

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

System.string 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

Sep 29, 2015

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 a 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

Sep 29, 2015

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 the 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 a 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

Sep 29, 2015

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

System.string 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

Sep 29, 2015

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

Sep 29, 2015

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

-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 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

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 login (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 login (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 login (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

Sep 29, 2015

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

Sep 29, 2015

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.XDInterServiceTypes.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

Sep 29, 2015

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

Sep 29, 2015

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
```

```
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-HypService

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Systemstring You can pipe a string that contains a path to Get-HypConfigurationDataForItem

Return Values

Systemstring.

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015
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\{<hostingUnitUid>}\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

Sep 29, 2015

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 Uid>

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

Sep 29, 2015

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[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-HypHypervisorConnectionMetadata [-InputObject] <HypervisorConnection[]> -Name <String> [-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

Sep 29, 2015
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

Sep 29, 2015

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

System.string 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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015
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 XDHy 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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015
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> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-HypHostingUnitMetadata [-HostingUnitName] <String> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress
<String>] [<CommonParameters>]

Set-HypHostingUnitMetadata [-InputObject] <HostingUnit[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]
[<CommonParameters>]

Set-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.

Related topics

- [Add-HypHostingUnitMetadata](#)
- [Remove-HypHostingUnitMetadata](#)

Parameters

-HostingUnitUid<Guid>

Id of the HostingUnit

Required?	true
Default Value	
Accept Pipeline Input?	true (ByValue, ByPropertyName)

-HostingUnit Name<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

Sep 29, 2015
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

System.string 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.

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>Set-HypHostingUnitStorage -LiteralPath XDHyp:\HostingUnits\MyHostingUnit -StoragePath 'XDHyp:\HostingUnits\MyHostingUnits\newStorage.storage' -Superseded $true
```

```
HostingUnitId      : 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

Sep 29, 2015
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

Sep 29, 2015

Adds or updates metadata on the given Service.

Syntax

```
Set-HypServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-HypServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-HypServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-HypServiceMetadata [-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-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)

-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?	

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

Sep 29, 2015
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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

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-HypHypervisorConnectionNameAvailable

Sep 29, 2015

Checks to ensure that the proposed name for a hypervisor connection is unused.

Syntax

```
Test-HypHypervisorConnectionNameAvailable -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

Sep 29, 2015

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-HypHypervisorConnection -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

Sep 29, 2015

Overview

Name	Description
ProvMachineCreationSnapin	The Machine Creation Service PowerShell snap-in provides administrative
Prov Filtering	Describes the common filtering options for XenDesktop cmdlets.
providers	Describes how Windows PowerShell providers provide access to data and

Cmdlets

Name	Description
Add-ProvSchemeControllerAddress	Adds a list of host names (as DNS addresses) to a provisioning scheme.
Add-ProvSchemeMetadata	Adds metadata on the given ProvisioningScheme.
Add-ProvSchemeScope	Add the specified ProvisioningScheme(s) to the given scope(s).
Add-ProvTaskMetadata	Adds metadata on the given Task.
Get-ProvDBConnection	Gets the database string for the specified data store used by the MachineCreation Service.
Get-ProvDBSchema	Gets a script that creates the MachineCreation Service database schema for the specified data store.
Get-ProvDBVersionChangeScript	Gets a script that updates the MachineCreation Service database schema.
Get-ProvInstalledDBVersion	Gets a list of all available database schema versions for the MachineCreation Service.
Get-ProvObjectReference	Returns the number of local objects holding references to objects from other services.
Get-ProvScheme	Gets the list of provisioning schemes.
Get-ProvSchemeMasterVMImageHistory	Gets the list of master VM snapshots that have been used to provide hard disks to provisioning schemes.

Name	Description
Get-ProvScopedObject	Gets 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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015
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?	

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

-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 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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

Gets the details of the scoped objects for the MachineCreation Service.

Syntax

```
Get-ProvScopedObject [-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_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

Objectld : cd4174ee-9e4b-4e57-b126-9dbf757fe493

ObjectName : MyExampleScheme

Description : Test scheme

ScopelId : 304e0fa7-d390-47f0-a94f-7e956a324c41

ScopeName : Finance

ObjectType : Scheme

Objectld : cd4174ee-9e4b-4e57-b126-9dbf757fe493

ObjectName : MyExampleScheme

Description : Test scheme

ScopelId :

ScopeName :

ObjectType : Scheme

Objectld : 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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015
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.

MasterVmlImageIsASnapshot

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 -IdentityPoolName 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.snapshot
```

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-db85ed9d351b
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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015

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[]> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Remove-ProvSchemeMetadata [-InputObject] <ProvisioningScheme[]> -Map <PSObject> [-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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

Removes metadata from the given Service.

Syntax

```
Remove-ProvServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-ProvServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-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)

-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-ProvService | % { Remove-ProvServiceMetadata -Map $_.MetadataMap }
```

Remove all metadata from all Service objects.

Remove-ProvTask

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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	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

Sep 29, 2015
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

Sep 29, 2015

Changes the parameter values for a provisioning scheme.

Syntax

```
Set-ProvScheme [-ProvisioningSchemeName] <String> -IdentityPoolName <String> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme [-ProvisioningSchemeName] <String> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme [-ProvisioningSchemeName] <String> -IdentityPoolUid <Guid> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme -ProvisioningSchemeUid <Guid> -IdentityPoolUid <Guid> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme -ProvisioningSchemeUid <Guid> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-PassThru] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

```
Set-ProvScheme -ProvisioningSchemeUid <Guid> -IdentityPoolName <String> [-VMCpuCount <Int32>] [-VMMemoryMB <Int32>] [-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

-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

-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

-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

-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?	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

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

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

Sep 29, 2015
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> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]  
  
Set-ProvSchemeMetadata [-ProvisioningSchemeName] <String> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]  
  
Set-ProvSchemeMetadata [-InputObject] <ProvisioningScheme[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]  
  
Set-ProvSchemeMetadata [-InputObject] <ProvisioningScheme[]> -Map <PSObject> [-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<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 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

Sep 29, 2015

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

Sep 29, 2015

Adds or updates metadata on the given Service.

Syntax

```
Set-ProvServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ProvServiceMetadata [-ServiceHostId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-ProvServiceMetadata [-InputObject] <Service[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress  
<String>] [<CommonParameters>]
```

```
Set-ProvServiceMetadata [-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-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)

-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 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

Sep 29, 2015

Adds or updates metadata on the given Task.

Syntax

```
Set-ProvTaskMetadata [-TaskId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ProvTaskMetadata [-TaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ProvTaskMetadata [-InputObject] <Task[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-ProvTaskMetadata [-InputObject] <Task[]> -Map <PSObject> [-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)

-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 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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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-ProvDBConnection -DBConnection "Server=serverName\SQLEXPRESS;Initial Catalog = databaseName; Integrated Security = True"
```

OK

Tests a database connection string for the MachineCreation Service.

----- **EXAMPLE 2** -----

```
c:\PS>Test-ProvDBConnection -DBConnection "Invalid Connection String"
```

Invalid database connection string format.

Tests an invalid database connection string for the MachineCreation Service.

Test-ProvSchemeNameAvailable

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
Overview

Name	Description
MonitorMonitorSnapin	The Monitor Service PowerShell snap-in provides administrative functions for the Monitor Service.
Monitor Filtering	Describes the common filtering options for XenDesktop cmdlets.

Cmdlets

Name	Description
Get-MonitorConfiguration	Gets the configuration settings currently being used by the Monitor Service.
Get-MonitorDataStore	Gets details for each of the Monitor data stores.
Get-MonitorDBConnection	Gets the database string for the specified data store used by the Monitor Service.
Get-MonitorDBSchema	Gets a script that creates the Monitor Service database schema for the specified data store.
Get-MonitorDBVersionChangeScript	Gets a script that updates the Monitor Service database schema.
Get-MonitorInstalledDBVersion	Gets a list of all available database schema versions for the Monitor Service.
Get-MonitorService	Gets the service record entries for the Monitor Service.
Get-MonitorServiceAddedCapability	Gets any added capabilities for the Monitor Service on the controller.
Get-MonitorServiceInstance	Gets the service instance entries for the Monitor Service.
Get-MonitorServiceStatus	Gets the current status of the Monitor Service on the controller.
Remove-MonitorServiceMetadata	Removes metadata from the given Service.
Reset-MonitorDataStore	Refreshes the database string currently being used by the Monitor service.

Name	Description
Reset-MonitorServiceGroupMembership	Resets the access permissions and configuration service locations for the Monitor Service.
Set-MonitorConfiguration	Sets the database string currently being used by the Monitor Service.
Set-MonitorDBConnection	Configures a database connection for the Monitor Service.
Set-MonitorServiceMetadata	Adds or updates metadata on the given Service.
Test-MonitorDBConnection	Tests a database connection for the Monitor Service.

about_MonitorMonitorSnapin

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-MonitorServiceMetadata [-InputObject] <Service[]> -Name <String> [-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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

Sets the database string currently being used by the Monitor Service.

Syntax

```
Set-MonitorConfiguration [-GroomSessionsRetentionDays <Int32>] [-GroomFailuresRetentionDays <Int32>] [-GroomLoadIndexesRetentionDays <Int32>] [-GroomDeletedRetentionDays <Int32>] [-GroomSummariesRetentionDays <Int32>] [-DataCollectionEnabled <Boolean>] [-FullPollStartHour <Int32>] [-ResolutionPollTimeHours <Int32>] [-SyncPollTimeHours <Int32>] [-DetailedSqlOutputEnabled <Boolean>] [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]
```

Detailed Description

Sets the string for the database connection currently being used by the Monitor Service.

A database connection need not be configured for this command to be used.

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

-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

-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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
Overview

Name	Description
SfStorefrontSnapin	This service short description
Sf Filtering	Describes the common filtering options for XenDesktop cmdlets.

Cmdlets

Name	Description
Add-SfServerToCluster	Adds a new server to an existing cluster.
Add-SfStorefrontAddress	Modifies a StoreFront address configuration by adding an additional StoreFront address to it.
Get-SfCluster	Gets all Storefront clusters present in the site.
Get-SfDBConnection	Gets the database string for the specified data store used by the Storefront Service.
Get-SfDBSchema	Gets a script that creates the Storefront Service database schema for the specified data store.
Get-SfDBVersionChangeScript	Gets a script that updates the Storefront Service database schema.
Get-SfInstalledDBVersion	Gets a list of all available database schema versions for the Storefront Service.
Get-SfIsStorefrontInstalled	Tells whether StoreFront Services and Privileged Service are installed.
Get-SfService	Gets the service record entries for the Storefront Service.
Get-SfServiceAddedCapability	Gets any added capabilities for the Storefront Service on the controller.
Get-SfServiceInstance	Gets the service instance entries for the Storefront Service.
Get-SfServiceStatus	Gets the current status of the Storefront Service on the controller.

Name	Description
Get-SfStorefrontAddress	Sets the high-level description of a configuration for StoreFront addresses, based on a configuration byte array.
Get-SfTask	Gets the task history for the Storefront Service.
New-SfCluster	Creates new Storefront cluster with default set of services.
New-SfStorefrontAddress	Creates a new StoreFront address configuration, specifying a single address.
Remove-SfServerFromCluster	Removes server from the cluster.
Remove-SfServiceMetadata	Removes metadata from the given service.
Remove-SfTask	Removes from the database completed tasks for the Storefront Service.
Remove-SfTaskMetadata	Removes metadata from the given Task.
Reset-SfServiceGroupMembership	Reloads the access permissions and configuration service locations for the Storefront Service.
Set-SfCluster	Sets the parameters on the given cluster.
Set-SfDBConnection	Configures a database connection for the Storefront Service.
Set-SfServiceMetadata	Adds or updates metadata on the given service.
Set-SfTaskMetadata	Adds or updates metadata on the given Task.
Test-SfDBConnection	Tests a database connection for the Storefront Service.

about_SfStorefrontSnapin

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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 needed 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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

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-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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015

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

Sep 29, 2015
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

Sep 29, 2015

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

Sep 29, 2015

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-functionaly 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 needed to be http or https, be absolute and share the same schema and port.

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

-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

Sep 29, 2015
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

Sep 29, 2015

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 needed 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

Sep 29, 2015

Removes metadata from the given service.

Syntax

```
Remove-SfServiceMetadata [-ServiceHostId] <Guid> -Name <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfServiceMetadata [-ServiceHostId] <Guid> -Map <PSObject> [-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)

-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-SfService | % { Remove-SfServiceMetadata -Map $_.MetadataMap }
```

Remove all metadata from all service objects.

Remove-SfTask

Sep 29, 2015

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

Sep 29, 2015

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[]> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Remove-SfTaskMetadata [-InputObject] <Task[]> -Name <String> [-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

Sep 29, 2015
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-StServiceGroupmembership
```

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

Sep 29, 2015

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 needed 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

Sep 29, 2015
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

Sep 29, 2015

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?	

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

Sep 29, 2015

Adds or updates metadata on the given Task.

Syntax

```
Set-SfTaskMetadata [-TaskId] <Guid> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-SfTaskMetadata [-TaskId] <Guid> -Map <PSObject> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-SfTaskMetadata [-InputObject] <Task[]> -Name <String> -Value <String> [-LoggingId <Guid>] [-AdminAddress <String>]  
[<CommonParameters>]
```

```
Set-SfTaskMetadata [-InputObject] <Task[]> -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-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)

-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-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

Sep 29, 2015
Tests a database connection for the Storefront Service.

Test-SfDBConnection [-DBConnection] <String> [-LoggingId <Guid>] [-AdminAddress <String>] [<CommonParameters>]

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.

[Get-SfServiceStatus](#)

[Get-SfDBConnection](#)

[Set-SfDBConnection](#)

-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

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.

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.

----- 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.

Hinweise zu Drittanbietern

Dec 07, 2015

XenApp und XenDesktop 7.5 enthalten ggf. Software von Drittanbietern, die gemäß den in den folgenden Dokumenten aufgeführten Bestimmungen lizenziert ist:

- [XenApp 7.0, 7.1, and 7.5 and XenDesktop 7.0, 7.1, and 7.5 Third Party Notices](#)
- [FLEXnet Publisher Documentation Supplement: Software Licenses](#)